EXHIBIT 3

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

CORRIGENT CORPORATION, Plaintiff, v. DELL TECHNOLOGIES INC. and DELL INC., Defendants.	C.A. No. 22-496 (RGA) **JURY TRIAL DEMANDED**
CORRIGENT CORPORATION, Plaintiff, v. ARISTA NETWORKS, INC., Defendant.	C.A. No. 22-497 (RGA) **JURY TRIAL DEMANDED**

DECLARATION OF ANDREW C. MAYO IN SUPPORT OF CORRIGENT CORPORATION'S REPLY CLAIM CONSTRUCTION BRIEF

I, Andrew C. Mayo, hereby declare as follows:

1. I am a partner with the law firm of Ashby & Geddes, counsel of record for Plaintiff Corrigent Corporation in this matter. I make this declaration from personal knowledge and, if called to testify, I could and would testify competently thereto.

2. Attached hereto as **Exhibit 3A** is a true and correct copy of *Dell Technologies Inc. et al v. Corrigent Corporation*, IPR2023-00370, Petition, Paper 1, dated December 23, 2022.

3. Attached hereto as **Exhibit 3B** is a true and correct copy of *Dell Technologies Inc. et al v. Corrigent Corporation*, IPR2023-00370, Exhibit 1003, Declaration of Dr. Bambos, dated December 23, 2022.

4. Attached hereto as **Exhibit 3C** is a true and correct copy of *Arista Networks, Inc., v. Corrigent Corporation*, IPR2023-00805, Petition, Paper 2, dated April 3, 2023.

5. Attached hereto as **Exhibit 3D** is a true and correct copy of *Arista Networks, Inc., v. Corrigent Corporation*, IPR2023-00805, Exhibit 1003, Declaration of Dr. Lavian, dated April 3, 2023.

6. Attached hereto as **Exhibit 3E** is a true and correct copy of an excerpt of the Ramesh Ponnapalli Deposition Transcript taken on October 11, 2023 in *Corrigent Corporation v. Cisco Systems, Inc.*, 6:22-cv-00396 (W.D. Tex.)

7. Attached hereto as **Exhibit 4A** is a true and correct copy of IEEE Std 802.1Q-1998, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, dated December 8, 1998.

8. Attached hereto as **Exhibit 4B** is a true and correct copy of Lasserre et al. *Virtual Private LAN Services over MPLS,* IETF draft-ietf-12vpn-vpls-1dp-08.txt, dated November 2005

9. Attached hereto as **Exhibit 4C** is a true and correct copy of DeCusatis, *Fiber Optic Data Communication Technological Trends and Advances*, dated 2002

10. Attached hereto as **Exhibit 4D** is a true and correct copy of RCA Engineer, Vol. 30, No. 1, dated Jan./Feb. 1985.

11. Attached hereto as **Exhibit 4E** is a true and correct copy of an excerpt of the Leon Bruckman Deposition Transcript taken on November 2, 2023 in *Corrigent Corporation v. Cisco Systems, Inc.*, 6:22-cv-00396 (W.D. Tex.).

12. Attached hereto as **Exhibit 4F** is a true and correct copy of IEEE Std. 896.1-1991, *IEEE Standard for Futurebus+ - Logical Protocol Specification*, September 26, 1991.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct. Executed on February 22, 2024 in Wilmington, Delaware.

/s/ Andrew C. Mayo
Andrew C. Mayo

EXHIBIT 3A

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DELL TECHNOLOGIES INC.,
DELL INC.,
and
CISCO SYSTEMS, INC.,
Petitioners

v.

CORRIGENT CORPORATION,
Patent Owner.

Case No. IPR2023-00370

U.S. Patent No. 7,593,400

Petition for *Inter Partes* Review of U.S. Patent No. 7,593,400

Mail Stop **PATENT BOARD**
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

CASES

REGULATIONS

PETITIONERS' EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 7,593,400 ("the '400 patent")
1002	Copy of Prosecution History of the '400 patent
1003	Declaration of Dr. Nicholas Bambos
1004	U.S. Patent Application Publication No. 2005/0198371 ("Smith")
1005	Certified English Translation of Japanese Patent Application No. 2005086668 ("Ishimori")
1006	U.S. Patent No. 6,735,198 ("Edsall")
1007	U.S. Patent Application Publication No. 2004/0133619 ("Zelig")
1008	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 1998 Edition ("802.1Q")
1009	Declaration of Dr. Mary K. Bolin
1010	*Corrigent Corp. v. Dell Technologies et al.*, No. 1:22-cv-00496 (D. Del.), Dkt. 1
1011	*Corrigent Corp. v. Arista Networks, Inc.*, No. 1:22-cv-00497 (D. Del.), Dkt. 1
1012	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 1
1013	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 42
1014	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 46
1015	IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition
1016	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition
1017	IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition

Exhibit No.	Description
1018	Kompella et al., *Virtual Private LAN Service*, IETF (December 2005)
1019	Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005)
1020	Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005)
1021	U.S. Patent No. 6,917,986
1022	U.S. Patent No. 7,974,223
1023	Western District of Texas Statistics (Docket Navigator)
1024	District of Delaware Statistics (Docket Navigator)
1025	Japanese Patent Application No. 2005086668 (“Ishimori”)

I. INTRODUCTION

Dell Technologies Inc., Dell Inc., and Cisco Systems, Inc. hereby petition for *inter partes* review of U.S. Patent No. 7,593,400 and seek cancellation of claims 1–20 ("the Challenged Claims") because they are unpatentable under 35 U.S.C. § 103.

The '400 patent relates to communication networks, specifically synchronizing learned MAC addresses to bridge ports in a distributed or virtual bridge in a network. The patent relies on a known technique by which a group of parallel physical links between two endpoints in a network are joined together in what is called a link aggregation (LAG) group. *Id.*, 2:37–40. The LAG group is seen by the user as a single link. *Id.*, 2:31–33. The disclosed bridges also use a known concept called a MAC forwarding database (FDB) to store associations of MAC addresses and the connected ports. *Id.*, 3:7–11. When a line card—which receives and forwards packets within a bridge—receives a packet with a destination MAC address that appears in the FDB, it forwards the packet to the port associated with the MAC address in the FDB. *Id.* On the other hand, if the destination MAC address does not appear in the FDB, the line card floods the packet to all the ports. *Id.* This was all admittedly known in the art. *Id.*, 1:33–2:56.

The patent's alleged improvement is a synchronization method that purportedly avoids the need for excessive flooding when LAGs are used. According to the patent, when a data packet arrives at a line card, if the port of the line card on

which the packet was received belongs to a LAG group, the line card sends a synchronization message to the other line cards in the same LAG group. In this way, the patent suggests that other line cards in a LAG group can learn MAC addresses before they have received packets from the MAC address in question. *Id.*, 3:46–53.

However, this alleged advance is not new. To the contrary, both the problem and solution were disclosed in the prior art, as discussed herein. For these reasons, Petitioners respectfully request the Board institute IPR proceedings and cancel the Challenged Claims.

II. MANDATORY NOTICES

A. Real Parties-in-Interest

The real parties-in-interest are Dell Technologies Inc., Dell Inc., and Cisco Systems, Inc. (collectively, "Petitioners"). No unnamed entity is funding, controlling, or directing this Petition, or otherwise has had an opportunity to control or direct this Petition or Petitioners' participation in any resulting IPR.

B. Related Matters

The '400 patent is the subject of this IPR Petition is also the subject of patent litigation suits brought by Patent Owner Corrigent Corporation, including against Petitioners:

- *Corrigent Corp. v. Dell Technologies et al.*, No. 1:22-cv-00496 (D. Del.) (EX1010);

- *Corrigent Corp. v. Arista Networks, Inc.*, No. 1:22-cv-00497 (D. Del.) (EX1011); and
- *Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.) (EX1012).

C. Lead and Backup Counsel

Pursuant to 37 C.F.R. §§ 42.8(b)(3), 42.8(b)(4), and 42.10(a), Petitioners provide the following designation of counsel:

Lead Counsel	Back-up Counsel
Stuart Rosenberg (Reg. No. 60,772) Gibson, Dunn & Crutcher LLP 1881 Page Mill Road Palo Alto, CA 94304-1211 Tel: 650-849-5389 SRosenberg@gibsondunn.com	**First Back-Up Counsel** Brian Rosenthal (*pro hac vice* motion requested) Gibson, Dunn & Crutcher LLP 200 Park Avenue New York, NY 10166-0193 Tel: 212-351-2339 BARosenthal@gibsondunn.com **Additional Backup Counsel** Ryan Iwahashi (Reg. No. 63,378) Gibson, Dunn & Crutcher LLP 1881 Page Mill Road Palo Alto, CA 94304-1211 Tel: 650-849-5367 RIwahashi@gibsondunn.com

Petitioners respectfully request authorization to file a motion for Brian Rosenthal to appear before the USPTO *pro hac vice*. Mr. Rosenthal is an

experienced litigating attorney and is currently serving as lead counsel for Petitioners in related matters *Corrigent Corp. v. Dell Technologies et al.*, No. 1:22-cv-00496 (D. Del.) and *Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.). Mr. Rosenthal has established familiarity with the subject matter at issue in this proceeding. Petitioners intend to file a motion to appear *pro hac vice* under 37 C.F.R. § 42.10. Pursuant to 37 C.F.R. § 42.10(b), powers of attorney accompany this Petition.

D. Service Information

Service via hand delivery or postal mail may be made at the addresses of the lead and back-up counsel above. Petitioners hereby consent to electronic service, and service via electronic mail may be made at the email addresses provided above for the lead and back-up counsel.

III. PAYMENT OF FEES

Pursuant to 37 C.F.R. §§ 42.103 and 42.15(a), the required fee is being submitted herewith. The Office is authorized to charge any fee deficiency, or credit overpayment, to deposit account no. 50-1408. Any additional fees due in connection with this Petition may be charged to the foregoing account.

IV. REQUIREMENTS OF *INTER PARTES* REVIEW

A. Standing

Pursuant to 37 C.F.R. § 42.104(a), Petitioners certify that the ’400 patent is available for IPR and that no Petitioner is barred from requesting an IPR on the grounds identified in this Petition. Specifically, Petitioners certify that: Petitioners have not filed a civil action challenging the validity of the ’400 patent; this petition is filed not more than one year from April 19, 2022, the date on which the Petitioners were served with the complaint alleging infringement of the ’400 patent; the estoppel provisions of 35 U.S.C. § 315(e)(1) do not prohibit this IPR; and this petition is filed after the later of (a) the date that is nine months after the date of the grant of the ’400 patent or (b) the termination of any post-grant review of the ’400 patent.

B. Identification of Challenge and Relief Requested

Pursuant to 37 C.F.R. § 42.104(b), Petitioners request the Board institute IPR of claims 1–20 of the ’400 patent under pre-AIA 35 U.S.C. § 103 on the prior art references and grounds described below:

1. Smith

U.S. Patent Application Publication No. 2005/0198371 (“Smith”) (EX1004) qualifies as prior art under 35 U.S.C. § 102(a) and/or (e). Smith was filed on February 19, 2004 and published on September 8, 2005.

2. Ishimori

Japanese Patent Application No. 2005086668 ("Ishimori") (EX1005, EX1025) qualifies as prior art under 35 U.S.C. § 102(a) and/or (b). Ishimori was filed on September 10, 2003 and published on March 31, 2005.

3. Edsall

U.S. Patent No. 6,735,198 ("Edsall") (EX1006) qualifies as prior art under 35 U.S.C. § 102(a), (b), and/or (e). Edsall was filed on December 21, 1999 and issued on May 11, 2004.

4. Zelig

U.S. Patent Application Publication No. 2004/0133619 ("Zelig") (EX1007) qualifies as prior art under 35 U.S.C. § 102(a), (b), and/or (e). Zelig was filed on January 7, 2003 and published on July 8, 2004.

5. IEEE 802.1Q Standard

The IEEE 802.1Q-1998 Standard ("802.1Q") (EX1008) qualifies as prior art under 35 U.S.C. § 102(b). Specifically, 802.1Q was published on March 8, 1999 and was publicly available no later than December 20, 2001. EX1009 ¶¶17–28.

Smith, Ishimori, Zelig, and 802.1Q were not cited during the prosecution of the '400 patent. Edsall was cited during prosecution; however, Petitioners rely on Edsall only for disclosures of claim limitations that were found to be present in Edsall during prosecution—findings that the applicant did not challenge.

In this IPR, Petitioners apply the above references and assert the following grounds of rejection under 35 U.S.C. § 103:

Ground	Claims	Basis for Rejection[1]
1	1, 3, 6, 11, 13, 16	Obvious over Smith
2	1–3, 11–13	Obvious over Smith in view of Ishimori
3	1, 4–7, 10–11, 14–17, 20	Obvious over Smith in view of Ishimori in further view of Edsall
4	8–9, 18–19	Obvious over Smith in view of Ishimori in further view of Zelig
5	9, 19	Obvious over Smith in view of Ishimori in further view of Zelig and 802.1Q

C. How the Challenged Claims Are To Be Construed Under 37 C.F.R. § 42.104(b)(3)

A claim subject to *inter partes* review "shall be construed using the same claim construction standard that would be used to construe a claim in a civil action under 35 U.S.C. § 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent." 37 C.F.R. § 42.100(b) (November 13, 2018); 83 F.R. 51358 (2018). Petitioners do not believe

[1] All obviousness grounds include knowledge of a POSITA.

(Cont'd on next page)

any constructions have an impact on the invalidity analyses set forth herein.[2] *See HTC Corp. v. Cellular Commc'ns Equip., LLC*, 877 F.3d 1361, 1367–68 (Fed. Cir. 2017) (affirming the Board absent an express construction of a term).

D. How the Challenged Claims Are Unpatentable Under 37 C.F.R. § 42.104(b)(4)

The following sections explain how the Challenged Claims are unpatentable under the statutory grounds identified above, including where each element of the claim is found in the prior art patents or printed publications.

E. Supporting Evidence Under 37 C.F.R. § 42.104(b)(5)

The exhibit numbers of the supporting evidence relied upon and the relevance of the evidence to the Challenged Claims, including an identification of specific portions of the evidence that support the challenge, are provided below. The technical information and grounds for unpatentability are further supported by the Declaration of Dr. Nicholas Bambos (EX1003). A List of Exhibits is included in this paper pursuant to 37 C.F.R. § 42.63(e).

---

[2] Claim construction has begun in the *Cisco* litigation. Petitioners will request leave to submit the district court's claim construction order when available, so it is timely made of record and can be considered by the Board. 37 C.F.R. §42.100(b).

V. THRESHOLD REQUIREMENT FOR *INTER PARTES* REVIEW

Under 35 U.S.C. § 314(a), institution of *inter partes* review requires "a reasonable likelihood that the petitioners would prevail with respect to at least one of the claims challenged in the petition." This petition meets this threshold for each ground of unpatentability.

VI. PATENT OVERVIEW

A. Person of Ordinary Skill in the Art

The alleged invention relates to communication networks, specifically learning MAC addresses in a distributed or virtual bridge. EX1001, Abstract. A person of ordinary skill in the art ("POSITA") at the time of the alleged invention (May 2006) would have had a degree in electrical engineering or a similar discipline, with at least two years of relevant industry or research experience (or additional education). EX1003 ¶17. The relevant experience would include a working understanding of networking systems, including distributed bridges, MAC forwarding tables, LAG groups, and virtual LAN services. *Id.* Lack of professional experience can be remedied by additional education, and vice versa. *Id.*

B. Background in the Relevant Art

All concepts in this section were well known and widely used by POSITAs at least as of May 19, 2006. EX1003 ¶¶36–53.

1. Computer Networks

Computer networks are comprised of communications links that connect to ports of communication nodes. *Id.* ¶37. Data packets or "frames" transmit information between the nodes based on MAC (Media Access Control) addresses, which are unique identifiers assigned to network hardware. *Id.* ¶¶37,44–45. An L2 switching node maintains a MAC address table, which maps the output port for a particular frame based on its destination MAC address. *Id.* ¶¶47–50.

2. Switches and Learning MAC Tables

MAC addresses are used by L2 switches (also called bridges) to switch packets at the network node from an input port to an output port. *Id.* ¶¶44–46. A MAC bridge is a network device connects two physically separated LANs into a single logical LAN. *Id.* ¶44. To forward data packets, called "frames," a MAC bridge maintains a forwarding database (FDB) of MAC addresses and corresponding bridge ports of packets that have been received at the bridge. *Id.* ¶¶45–47. When a bridge receives a packet, it checks the destination MAC address of the packet. The bridge checks a MAC address table (or forwarding database) to determine whether it knows where to forward the message. *Id.* ¶47. If the destination MAC address of the packet is in the FDB, the bridge forwards the packet to the port associated with that destination MAC address. *Id.* If not, the bridge may send the message over every port to ensure it is received at the intended destination. *Id.* This is called

"flooding." *Id.* When packets are received, the bridge also "learns" information about the *source* MAC address of each packet and, if the source MAC address is not in the FDB, updating the database to reflect the association between the incoming port and the source MAC address. *Id.*

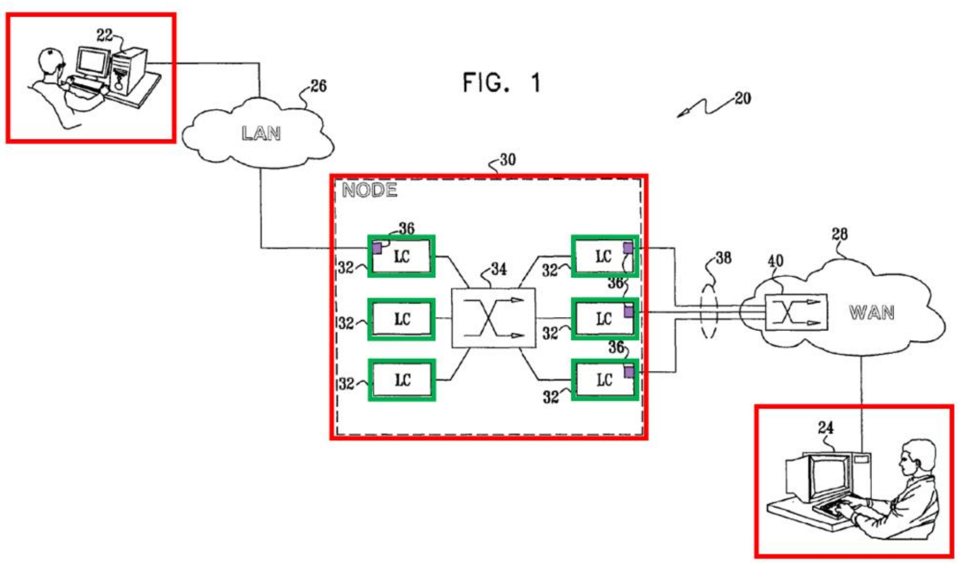
3. Link Aggregation

In the 1990s, link aggregation was used to increase bandwidth, traffic load balancing, scalability, and reliability by grouping multiple links from a switch to form a link aggregate group, or "LAG." *Id.* ¶51. These multiple links in a LAG are treated as a single logical link. *Id.* The Layer 2 Link Aggregation Sublayer combines a number of individual physical links into a single logical link, which presents a single MAC interface to the MAC client. *Id.* (citing EX1017, Section 3, Fig. 43-1).

C. Summary of the Alleged Invention of the '400 Patent

The '400 patent relates to communication networks, specifically learning MAC addresses in a distributed bridge. The patent acknowledges it was already well known that computing systems connect to LANs at OSI Layer 2. EX1001, 1:13–19. As the patent acknowledges, "a number of authors have described methods for creating a virtual private LAN service (VPLS), which links different LANs together over an IP network." *Id.*, 2:9–17. The patent explains a single provider can participate in multiple VPLSs. *Id.*, 2:18–30. Treating both virtual and physical

interfaces identically gives the illusion, from the user's perspective, that the provider network is a single LAN domain. *Id.*, 2:31–33. This VLAN functionality, which was known in the art, allows end points to be joined using link aggregation (LAG) as if they were part of the same local network even if they are not physically close to each other. *Id.*, 2:33–40 (citing EX1017). For example, as shown in Fig. 1, although users of **terminal 22 and 24** are not on the same LAN domain (terminal 22 is connected to a LAN 26 and terminal 24 is connected to a wide area network (WAN) 28), the VPLS "permits the users" to "communicate with one another as though they were connected to the same LAN domain." *Id.*, 5:51–62; Fig. 1.



A **network node 30** "comprises multiple **line cards 32**, linked by a switching core 34," and each line card has **ports 36**, which connect to other nodes in LAN 26 and WAN 28. *Id.*, 6:8–12. The line cards may serve as both ingress and egress with a MAC forwarding database (FDB), which stores learned MAC addresses and which

ports they were learned on. *Id.*, 3:7–11. When an ingress line card receives an incoming packet, it consults the FDB for the MAC destination address of the packet to determine which line card and port to forward the packet to. *Id.*, 3:11–17. When the MAC destination address does not appear in the FDB, the line card floods the packet to all of the ports, as in the prior art. *Id.* According to the patent, however, there was a need for better MAC address learning at the bridges within a VLAN to prevent the need to flood (or broadcast) messages when the destination MAC address is unknown. *Id.*, 3:34–53.

To address this, the purported invention provided "improved methods for MAC learning and network nodes that implement such methods." *Id.*, 2:60–62. It describes a synchronization method that allegedly avoids the need for constant flooding. *Id.*, 3:41–45. When the transmitting line card sends the data packet via a port that belongs to a LAG group, it sends a synchronization message so the other line cards in the same LAG group can learn the MAC address association even when those other line cards have not yet received packets from that MAC address. *Id.*, 3:46–53. Therefore, upon receiving an incoming packet, "[i]f the record is found, the packet processor adds a tag to the packet indicating the egress port through which the packet should be forwarded." *Id.*, 7:31–34. If there is "no corresponding record in the database however, it tags the packet for flooding." *Id.*, 7:42–44. While the "switching core 34 will pass the packet for transmission via all ports" that "are used

by this VPLS instance," "[f]or each LAG group serving the VPLS instance, however, the flooded packet is transmitted ***via only one port in the group***." *Id.*, 7:47–49 (emphasis added).

Additionally, these synchronization messages ("SYNC") are sent at regular intervals "to report each SELF entry [*e.g.*, entries that are learned by the packet processor on the line card itself] that it has created in the FDB 58 to the other line cards 32 in node 30." *Id.*, 8:17–22. The patent thus discloses a record is removed from the database "if a predetermined aging time elapses following the timestamp without a further packet having been received with the same key" and "free[s] up space for new records." *Id.*, 9:4–11. If, on the other hand, the current packet matches a record in the FDB, "the packet processor refreshes the timestamp of the record," and "forwards the packet to the appropriate output port." *Id.*, 9:29–31.

D. Summary of the Prosecution History of the '400 Patent

The '400 patent's application was filed on May 19, 2006. EX1001, [22]. The Examiner rejected all pending claims as anticipated by Edsall, or obvious over Edsall and U.S. Patent No. 6,788,681 ("Hurren"). EX1002, 81–85. During an Examiner interview, it was discussed if the limitation "link aggregation group having a plurality of ports" in claim 1 were amended to recite the claim term as defined by the specification at 2:37–40, then the amended claim would overcome Edsall. *Id.*, 112. The applicant therefore amended the claim limitation to add "a link aggregation

(LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards." *Id.*, 114. The applicant made no other amendments to overcome the rejections over Edsall. The Examiner allowed the claims, and the patent issued on September 22, 2009. *Id.*, 128, 160.

VII. GROUNDS OF UNPATENTABILITY

A. Ground 1: Claims 1, 3, 6, 11, 13, and 16 would have been obvious over Smith.

Smith renders obvious claims 1, 3, 6, 11, 13, and 16.

1. Overview of Ground 1

a) Smith

Smith is directed to a virtual network device comprising interface bundles, which are managed as a single logical interface. The virtual network devices contain multiple "sub-units," which "collectively operate as a single logical network device." EX1004, Abstract, ¶34. To achieve this, the "system includes a virtual link bundle" with "several communications links," and the "communications links are configured to be managed as a single link." *Id.* ¶9.

As depicted below, **virtual network device 202** is coupled to other network devices 120(1)–120(3). *Id.* ¶44. The virtual network device consists of virtual network device sub-units 122(1) and 122(2), which includes several **line cards**

304(1)–304(4). *Id.* ¶46. The line cards include a "forwarding engine" and "**interfaces**." *Id.* ¶¶46–47. When a packet is received at an uplink interface, the virtual network device sub-unit can learn "the sending device's [MAC] address" by "associating the MAC address with the logical identifier of the uplink interface." *Id.* ¶54. The sub-unit then informs each forwarding engine of this association. *Id.* "[P]ackets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier." *Id.*

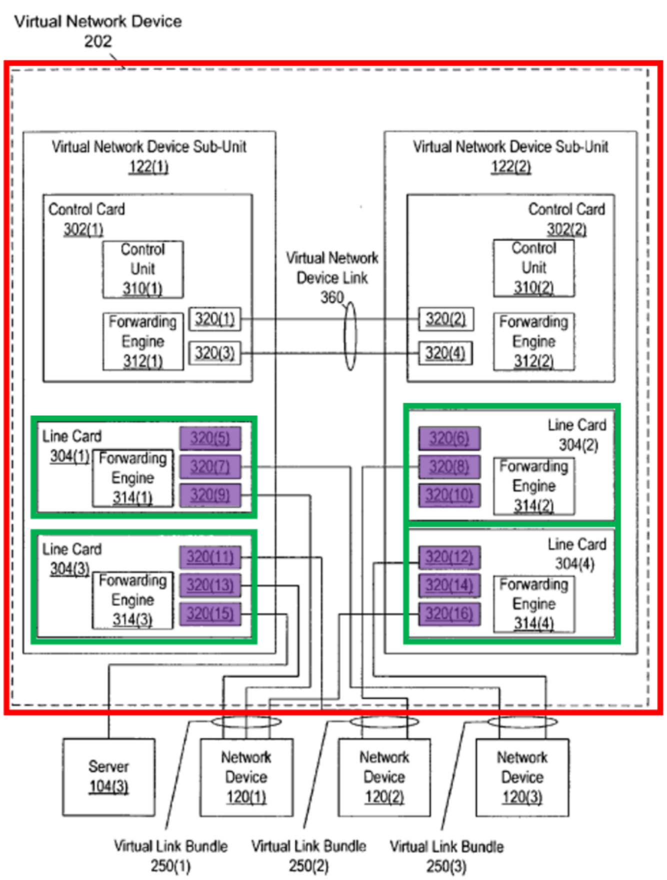


FIG. 3

Smith uses "MAC notification frames" in order to "keep the content of the L2 tables in the virtual network device sub-unit 122(1) synchronized with the content

of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶62. If a "forwarding table already includes an entry associating the destination address with a port of one of the network devices," that forwarding engine will generate "a MAC notification identifying this association" to "any other forwarding engines within" the virtual network device sub-unit. *Id.* ¶63. "If there is no hit in the forwarding table," "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN" excluding the "interface that the packet arrived on." *Id.* ¶66. As a result of the virtual link bundling, the data packet is sent via only "one of the communication links." *Id.* ¶9.

2. Analysis of Ground 1

a) Claim 1[pre]: A method for communication, comprising:

To the extent the preamble is limiting, Smith discloses claim 1[pre]. Smith's "system includes a virtual link bundle, which includes several communication links," which constitute methods for communication. *Id.* ¶9; EX1003 ¶98.

b) Claim 1[a]: configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;

Smith discloses claim 1[a]. Smith discloses a **virtual network device 202**, which is a network node. EX1004 ¶36. The virtual network device "includes **several cards**" such as 304(1) and 304(4) (*e.g.*, first and second line cards). *Id.* ¶46.

These line cards include **interfaces**, for example, 320(5), 320(7), and 320(9) on line card 304(1), and 320(12), 320(14), and 320(16) on line card 304(4) (*e.g.*, a plurality of ports). *Id.* ¶47. Thus, Smith's first and second line cards have respective first and second ports. EX1003 ¶99.

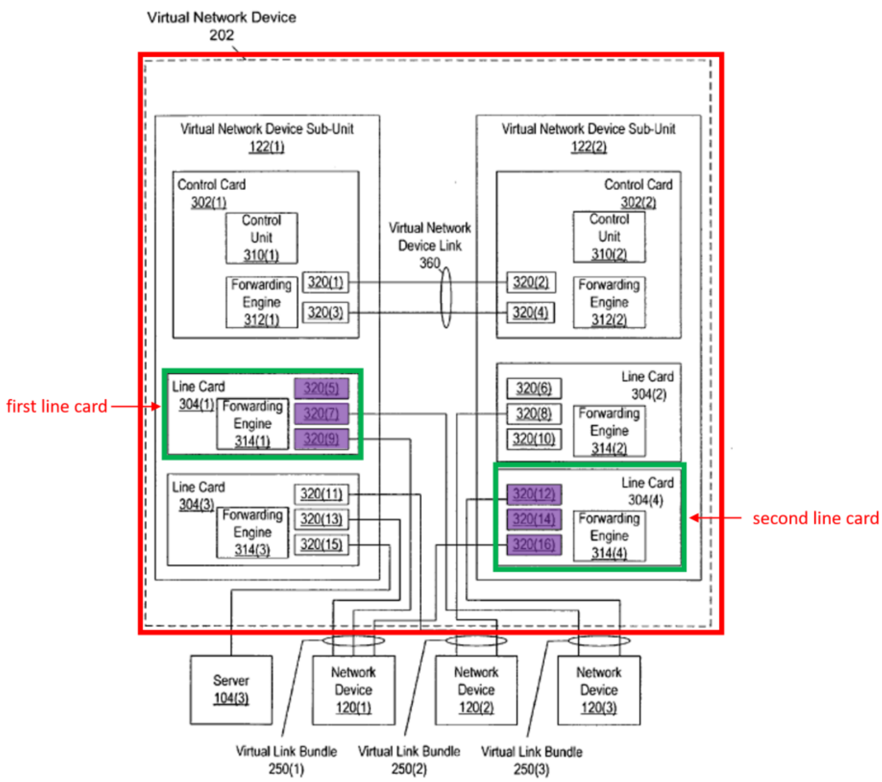


FIG. 3

An "interface" and "port" are synonymous and Smith treats them interchangeably, and thus, discloses the first and second ports. EX1004 ¶63 (stating "port *or* uplink interface"). Smith teaches the virtual link bundles, which are a link aggregation (LAG) group of parallel physical links, may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶6,9,30. The virtual network device 202 "route[s] and forward[s] packets to and from network devices 120(1)–

120(3)" by associating the MAC address of a received data packet with the logical identifier of the uplink interface. *Id.* ¶54. Smith's virtual network device 202 therefore operates as a distributed MAC bridge in a Layer 2 data network. EX1003 ¶¶100–102.

c) Claim 1[b]: configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined Member Line Cards;

Smith discloses claim 1[b]. As stated above, Smith's virtual link bundles, which are a LAG group of parallel physical links, may provide Layer 2 forwarding, and are "managed as a single link." EX1004 ¶¶6,9,30. Smith discloses **network device 120(2)** (*e.g.*, one endpoints) "is coupled to **virtual network device 202**" (*e.g.*, another endpoint) "by **virtual link bundle 250(2)**" (*e.g.*, a LAG group) as shown in the annotated figure below. *Id.* ¶44. The virtual link bundle 250(2) consists of **two uplinks** (*e.g.*, a plurality of LAG ports). *Id.* ¶51.

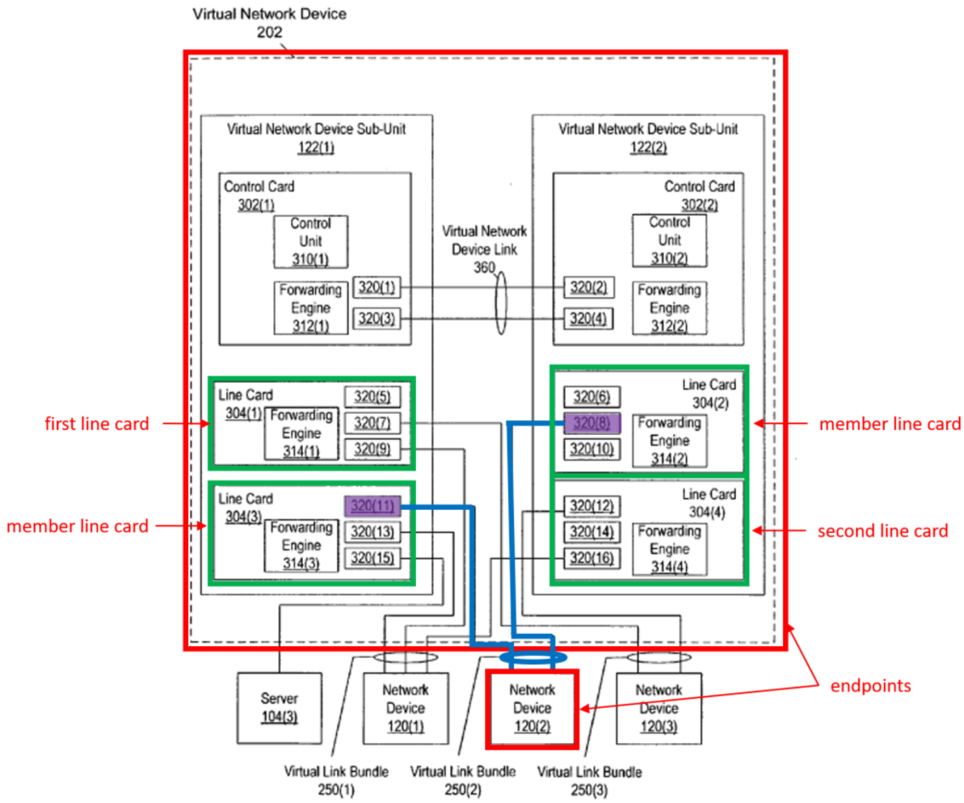


FIG. 3

Smith further specifies the virtual network devices provide Layer 2 "forwarding and routing." *Id.* ¶30. Because virtual network device sub-units 122(1) and 122(2) "can coordinate their behavior such that they appear to be a single virtual device," the remaining line cards 304(2) and 304(3) comprise the conjoined member line cards. Smith therefore discloses configuring a link aggregation (LAG) group of parallel physical links (*e.g.*, virtual link bundle 250(2)) between two endpoints (*e.g.*, virtual network device 202 and network device 120(2)) in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG

ports (*e.g.*, interfaces 320(8) and 320(11)) and a plurality of conjoined Member Line Cards (*e.g.*, line cards 304(2) and 304(3)). EX1003 ¶¶103–105.

d) Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

Smith discloses claim 1[c]. The member line cards in Smith include a **forwarding engine** and **interfaces** (*e.g.*, ports). *Id.* ¶47.

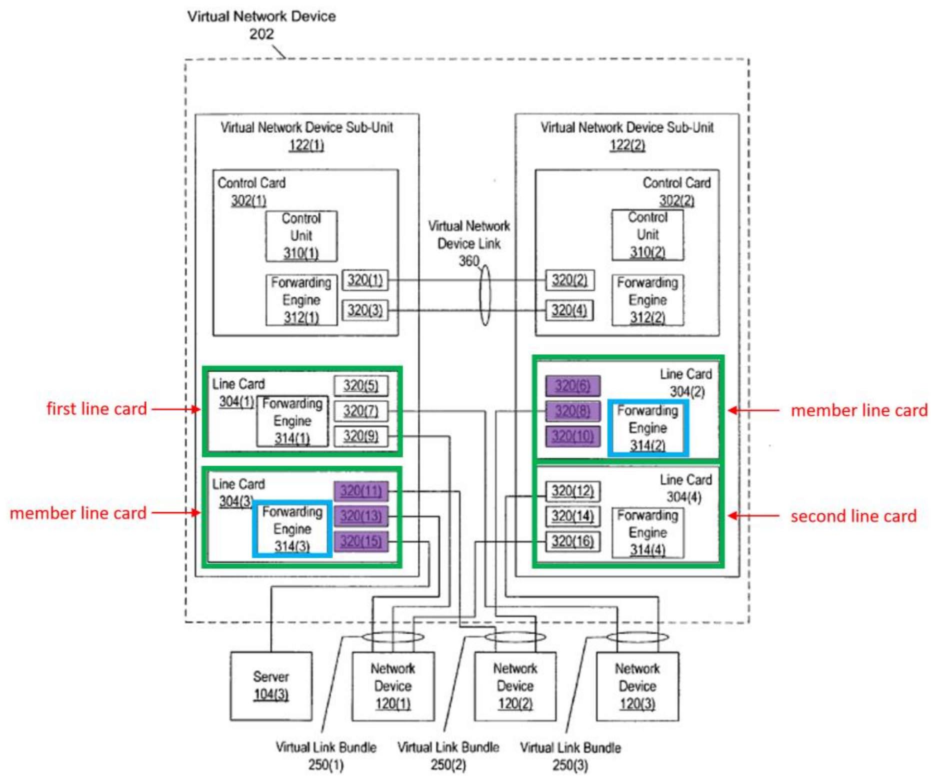


FIG. 3

When a packet is received on a particular uplink interface, the virtual network device learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] uplink interface" (*e.g.*, associating MAC addresses

with ports of said plurality of ports of said network node). *Id.* ¶54. The "forwarding engines" use the association between a packet and a particular logical identifier to route and forward packets to the network devices. *Id.* ¶54. Smith teaches this information may be used to "set up or modify lookup tables" (*e.g.*, the forwarding database (FDB)). *Id.* ¶¶57,61. Thus, Smith teaches the lookup table (*e.g.*, FDB) may store records associating MAC addresses with the logical identifier of the uplink interface (*e.g.*, ports of said plurality of ports) of said network node. EX1003 ¶¶106–107.

e) Claim 1[d]: receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;

Smith discloses claim 1[d]. Smith discloses the uplink interface (*e.g.*, ingress port) receives a data packet with the "sending device's MAC address" (*e.g.*, a MAC source address). EX1004 ¶54. Based on Figure 3, one such ingress port from a data packet received from network device 120(1) (*e.g.*, MAC source address) would be interface 320(9).

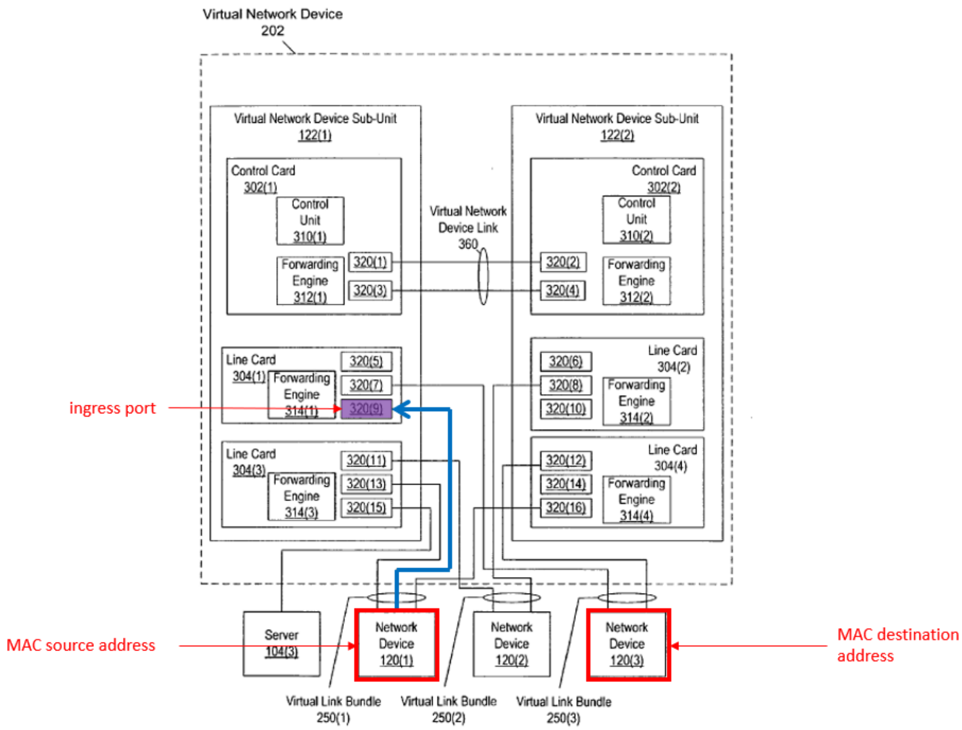


FIG. 3

Smith therefore teaches receiving a data packet on an ingress port (*e.g.*, interface 320(9)) of said network node (*e.g.*, virtual network device 202) from a MAC source address (*e.g.*, network device 120(1)). Smith further discloses this data packet has a "destination logical identifier," such as the MAC destination address of network device 120(3). *Id.* ¶¶60,62. Smith specifies these devices provide Layer 2 forwarding. *Id.* ¶30. Thus, Smith teaches this limitation. EX1003 ¶¶108–111.

f) Claim 1[e]: conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;

Smith discloses claim 1[e]. EX1003 ¶¶112–113. Smith teaches the data packet is forwarded based on "the association between a packet and a particular logical identifier." EX1004 ¶54. Smith additionally teaches its system "favor[s] local interfaces." *Id.* ¶56. Again, referring to Figure 3, in order to transmit the data packet to the MAC destination address (*e.g.*, network device 120(3)), the data packet would be conveyed from interface 320(9) to interface 320(7) (*e.g.*, said first port), because the system favors local interfaces. EX1003 ¶112. Interface 320(7), which is on line card 304(1) (*e.g.*, said first line card), would then transmit the data packet to the MAC destination address (*e.g.*, network device 120(3)).

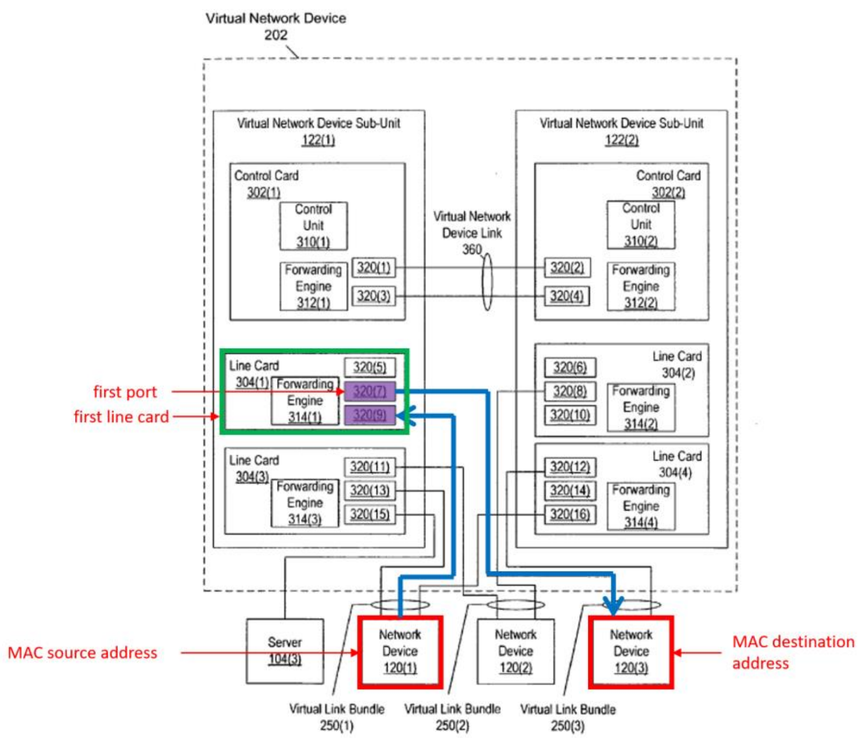


FIG. 3

g) Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one

and only one LAG port of said plurality of LAG ports;

Smith discloses claim 1[f]. EX1003 ¶¶114–116. Smith teaches the virtual network device sub-unit 122(1) looks up the destination address in a "lookup table" (*e.g.*, FDB). EX1004 ¶61. Claim 1[f] does not recite which forwarding database of the plurality of member line cards is checked.[3] The only FDBs previously recited in the claim are associated with the member line cards and the LAG ports. Smith discloses sharing entries of the FDB with all its member line cards, and therefore, it does not matter which FDB is checked. *Id.* ¶63. A POSITA would have understood that any one of Smith's FDBs could be checked, including the FDB associated with the member line cards. EX1003 ¶114. Another interpretation is the FDB that is checked is on the line card associated with the ingress port, which would be line card 304(1), continuing with the above example. *Id.* Although there is no line card or FDB associated with the ingress port recited in the claim, Smith would render

[3] Petitioners intend to argue in the parallel litigation "said FDB" is indefinite for lacking a clear antecedent basis. However, the claims are invalid as obvious under any interpretation. Regardless, the question of indefiniteness is not before the Board and is no barrier to the Board's inquiry. *Haas Automation, Inc. v. Olati LLC*, IPR2021-00146, Paper 11, 11–13 (P.T.A.B. May 18, 2021).

obvious this limitation under this interpretation as well. *Id.* Thus, under any possible interpretation of "said FDB," Smith renders obvious this limitation. *Id.*

If there is no hit in the forwarding table, "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN." EX1004 ¶¶63,66. Smith expressly discloses or at least renders obvious that "flooding" is "via one and only one LAG port." In particular, Smith teaches when the network device sends a packet "via the virtual link bundle," it "selects one of the communication links on which to send the packet." *Id.* ¶9; EX1003 ¶116.

h) Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

Smith discloses claim 1[g]. Smith teaches the virtual network device sub-unit learns the source identifier of the sending device (*e.g.*, MAC source address). EX1004 ¶65. These identifiers are stored in a lookup table (*e.g.*, FDB) on a virtual network device. *Id.* ¶61. It would have therefore been obvious that the MAC source address (*e.g.*, network device 120(1)) would have had to be checked in the records of the FDB of said first line card (*e.g.*, line card 304(1)) in order to determine whether it existed in the FDB. EX1003 ¶¶117–118.

i) Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the

association to each member line card and said plurality of member line cards.

Smith discloses claim 1[h]. EX1003 ¶¶119–120. Smith teaches the virtual network device sub-unit learns the source identifier of the sending device (*e.g.*, MAC source address). EX1004 ¶65. Smith then teaches the network device sub-unit 122(2) sends a MAC notification (*e.g.*, a message) to update the forwarding engines (*e.g.*, sending a message of the association to each member line card) when it learns of a new association. *Id.* ¶63. "After being updated based on the MAC notification," the forwarding engines "now know the location of the device identified by the destination address." *Id.* A POSITA would have understood the MAC notification (*e.g.*, message) is therefore sent to each member line card. EX1003 ¶120. A POSITA would have also understood knowing the association between the MAC source address and ingress port requires adding an entry in the lookup table of, for example, line card 304(1) (*e.g.*, said first line card). *Id.*

j) Claim 3: The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

Smith discloses claim 3. Smith discloses the network device sub-unit 122(2) sends a MAC notification (*e.g.*, the message) to update the forwarding engines. EX1004 ¶63. Smith specifies "[a]fter being updated based on the MAC notification,

the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* A POSITA would understand this includes the other line cards in virtual network device 202, such as line card 304(4). EX1003 ¶122. For example, line card 304(4) (*e.g.*, the second line card) would receive the MAC notification (*e.g.*, the message) and update its lookup tables. EX1004 ¶57. Thus, Smith discloses in response to the message, the record of association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card. EX1003 ¶¶121–122.

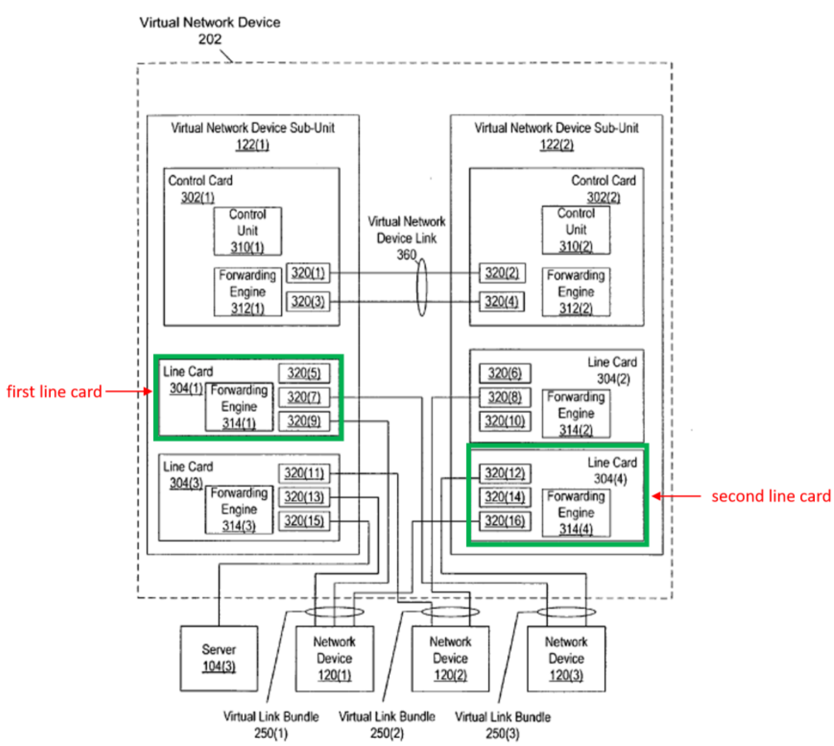


FIG. 3

> **k) Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via**

switching core of the network node to at least the second line card.

Smith discloses claim 6. The MAC notification frames are used to keep the content of the L2 tables synchronized. *Id.* ¶62. Thus, it would have been obvious to a POSITA to implement the MAC notification frame as a synchronization packet and transmit it to at least the second line card via the switching core. EX1003 ¶¶123–125.

l) Claim 11[pre] 11: A node for network communication, comprising:

To the extent the preamble is limiting, Smith discloses claim 11[pre]. Smith's system contains communications links that are coupled to network devices, which is a node. EX1004 ¶9; EX1003 ¶126.

m) Claim 11[a]: a switching core;

Smith discloses claim 11[a]. A POSITA would have understood that line cards need a way to communicate with each other. EX1003 ¶127. It would have been obvious to POSITA to do so via a switching core. *Id.*

n) Claim 11[b]: a plurality of ports;

Smith discloses claim 11[b]. *Id.* ¶128. The network devices in Smith contain several ports (*e.g.*, a plurality of ports). EX1004 ¶6 ("EtherChannel (TM) port bundle can be formed ***from several ports on a switch***….").

o) Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of

parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

Smith discloses claim 11[c].[4] §§VII.A.2.b)–d),VII.A.2.m); EX1003 ¶129.

p) Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address

[4] Petitioners also intend to argue "virtual MAC bridge" is indefinite. However, the claims are invalid as obvious under any interpretation. The question of indefiniteness is not before the Board and is no barrier to the Board's inquiry. *Haas*, Paper 11, 11–13.

does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

Smith discloses claim 11[d]. §§VII.A.2.e)–i); EX1003 ¶130.

q) Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.

Smith discloses claim 13. §VII.A.2.j); EX1003 ¶131.

r) Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

Smith discloses claim 16. §VII.A.2.k); EX1003 ¶132.

B. Ground 2: Claims 1–3 and 11–13 would have been obvious over Smith in view of Ishimori.

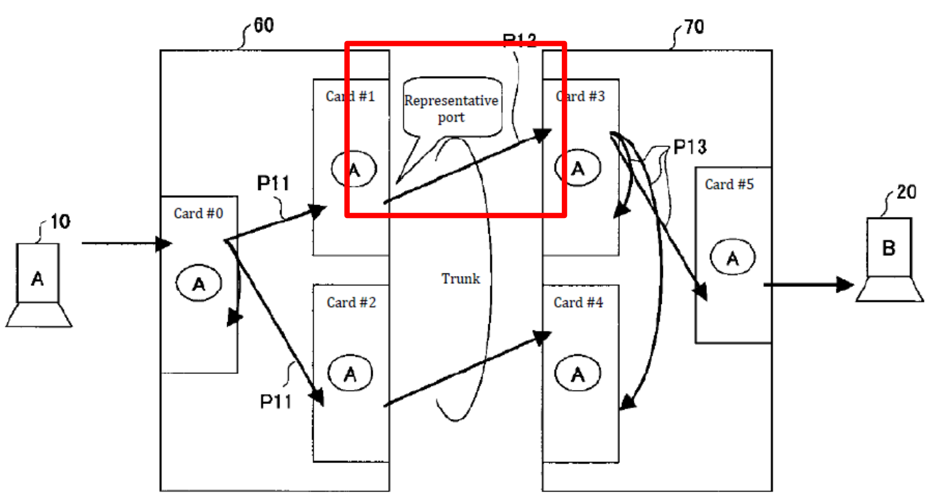
Although Smith renders obvious claims 1, 3, 11, and 13 alone (as described above), to the extent any elements are not disclosed or rendered obvious by Smith, they would have been obvious in combination with Ishimori. Petitioners incorporate by reference the analysis of Smith's disclosures from above. §VII.A.2. Additionally, the combination of Smith and Ishimori renders obvious claims 2 and 12.

1. Overview of Ground 2

a) Ishimori

Ishimori is directed to a packet forwarding method that uses "path learning" and "link aggregation" to "prevent repeated flooding." EX1005, Abstract.

Ishimori's system has node groups 60, 70 in between two terminals A10 and B20. *Id.*, Fig. 9. The node groups 60, 70 have communication cards #0–2 and #3–5, respectively. *Id.* ¶15. Ishimori teaches a method where the source MAC address of a received packet is learned by storing the source MAC address "in a buffer [*e.g.*, MAC table] had by each node." *Id.* ¶2. Thus, the destination MAC address for a packet is searched for in the MAC table of the local device. *Id.* ¶4. If the result is found in the buffer (*e.g.*, MAC table), then the packet is transmitted according to the forwarding path that was learned. *Id.* Thus, "as long as a learning result relating to the destination MAC address (DA)" exists in the buffer, "the forwarding path to apply to the corresponding packet can be decided unambiguously by using this information," and "there is no need for flooding." *Id.*



[FIG. 9]
Block diagram for describing one example of conventional link aggregation function

However, if the MAC address is not found, Ishimori acknowledges the inefficiency of flooding to all nodes, because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶2,19. Ishimori's solution to this flooding problem is the same solution the '400 patent later claimed. Specifically, Ishimori teaches, by leveraging "link aggregation," which bundles a plurality of ports to function as one virtual port, that "one representative port is selected from among the large number of ports." *Id.* ¶13. And the packet is therefore only flooded to "this **representative port**." *Id.* "[A] packet received from the terminal 10 and addressed to terminal 20 is received at card #0" of node 60. *Id.* ¶15. "One representative port is selected" and "this packet is further transmitted." *Id.* Additionally, "each communication card #0 to #5 of the node groups 60, 70" learns "information relating to the path leading to the location device" in "association with the source address." *Id.* ¶16. Ishimori teaches a learning method that "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." *Id.* ¶25. The "learn packet," which informs the plurality of line cards regarding new associations, "is transmitted to all nodes having the corresponding trunk." *Id.* ¶33.

Ishimori also teaches an aging process. When the buffers of the cards of each node learns a MAC address, the "hit bit" corresponding to the learned MAC address is set to "1." *Id.* ¶9. Then, after the first aging cycle, the hit bit is set to "0," which

indicates the MAC address should be deleted from the buffer on the next cycle. If the MAC address is learned again at the next aging process, the hit bit is reset to "1." *Id.* This prevents "performing flooding repeatedly and thus inviting increased line traffic." *Id.* ¶25.

b) Motivation to combine Smith and Ishimori

A POSITA would have been motivated to combine Smith and Ishimori, because both references disclose link bundling and path learning techniques in communication networks. As explained above, Smith discloses sending a packet to only "one of the communication links." EX1004 ¶9. Smith further teaches for interfaces included in interface bundles, "the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet." *Id.* ¶66. The purpose of flooding over only one link in a bundle is to prevent sending packets over unnecessary links. EX1003 ¶¶85–87. For example, a POSITA would have understood flooding would not occur over the link from which the packet was received. *Id.* ¶87.

Similarly, to the extent Smith does not explicitly disclose flooding via only one port in a LAG, Ishimori discloses another way of decreasing the number of links over which flooding occurs by explaining that flooding in a LAG need only happen over one representative port. EX1005 ¶13. Ishimori recognized the shortcomings in the approach of flooding all ports because the learning results on the

communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶19. Ishimori explains using "one representative port" prevents the application of "a different path in the same trunk to the same packet." *Id.* ¶¶13–14. A POSITA looking to solve the shortcomings of flooding all the links would have therefore looked to Smith and Ishimori. EX1003 ¶¶85–87. Smith discloses the concept of selecting one egress interface on a virtual network bundle, and Ishimori provides using "one representative port" prevents constant flooding.

Additionally, Ishimori teaches a "learn packet" is generated at a "predetermined timing." EX1005 ¶25. A POSITA looking to limit flooding over unnecessary links would have therefore been motivated also to implement Ishimori's learn packet generated at a predetermined timing into Smith's MAC notification method. EX1003 ¶88. This would have been within the knowledge of a POSA implementing such a method to make this modification and combine the teachings of Smith and Ishimori and have a reasonable expectation of success. EX1003 ¶¶85–88.

2. Analysis of Ground 2

a) Claim 1[c]

The combination of Smith and Ishimori renders obvious claim 1[c]. Smith discloses this limitation. §VII.A.2.d). To the extent this limitation is not disclosed by Smith, it is disclosed by the combination of Smith and Ishimori. Each of

Ishimori's communication cards has a "MAC table." EX1005 ¶2. Ishimori explains when a packet from terminal A is received at port #0 of card #0, the path information for address "A" is learned by storing the information in the MAC table "had by this card" (*e.g.*, card #0). *Id.* ¶3. Ishimori's line cards therefore maintain a MAC table (*e.g.*, FDB) to store path information between MAC address and the receiving card and port (*e.g.*, hold records associating MAC addresses with ports of said plurality of ports of said network node) *Id.* ¶3; EX1003 ¶¶134–136. It would have been obvious to substitute the line cards from Ishimori into Smith, and a POSITA would have had a reasonable expectation of success in making this simple substitution. *Id.* ¶136.

b) Claim 1[f]

The combination of Smith and Ishimori render obvious claim 1[f]. Smith discloses this limitation. §VII.A.2.g). To the extent this limitation is not disclosed by Smith, it is disclosed by the combination of Smith and Ishimori. Ishimori expressly discloses flooding is performed "when an address is not learned" on "one representative port" (*e.g.*, via one and only one LAG port). EX1005 ¶13. A POSITA would have been motivated to combine Smith with Ishimori because Ishimori explains using "one representative port" prevents "the application of a different path in the same trunk to the same packet." *Id.* ¶¶13,14; EX1003 ¶¶137–138. Thus, a POSITA would have looked to both Smith and Ishimori, which explain why it would

be beneficial to flood only one representative port, to solve the shortcomings of flooding all ports. EX1003 ¶¶87,138. Furthermore, a POSITA would have had a reasonable expectation of success in combining Smith with Ishimori because it would have been a simple application of Ishimori's methods on Smith's virtual network device. *Id.* ¶88,138.

c) Claim 1[g]

The combination of Smith and Ishimori renders obvious claim 1[g]. Smith discloses this limitation. §VII.A.2.h). To the extent this limitation is not disclosed by Smith, it is disclosed by the combination of Smith and Ishimori. Ishimori teaches the source MAC address is stored with the reception path information including the card number and port number. EX1005 ¶3. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.*; EX1003 ¶¶139–140.

d) Claim 1[h]

The combination of Smith and Ishimori render obvious claim 1[h]. Smith discloses this limitation. §VII.A.2.i). To the extent this limitation is not disclosed by Smith, it is disclosed by the combination of Smith and Ishimori. Ishimori discloses the source MAC address is stored with the reception path information

including the card number and port number. EX1005 ¶3. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.*

Ishimori further teaches a learning process where "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." *Id.* ¶25. Ishimori teaches the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶33. Ishimori therefore discloses sending a message of the association to each member line card for said plurality of member line cards. EX1003 ¶¶141–142. It would have been obvious to a POSITA to implement the MAC notification in Smith to perform the method taught in Ishimori to first check whether the MAC address is found in the MAC table, and if not, create a new record of the association. *Id.* ¶141. Both Smith and Ishimori teach a message of this association is sent to the plurality of member line cards. EX1004 ¶63; EX1005 ¶16.

e) Claim 2: The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC address and the respective ports.

The combination of Smith and Ishimori discloses claim 2. Smith teaches the network device sub-unit 122(2) sends a MAC notification (*e.g.*, a message) to update the forwarding engines, but it does not disclose doing so periodically at predefined times. EX1004 ¶63. However, Ishimori discloses this. Ishimori teaches a "learn packet" (*e.g.*, message) is generated at a "predetermined timing" (*e.g.*, periodically at predefined times). EX1005 ¶25, Fig. 15. Ishimori teaches the flooding operation informs the plurality of line cards regarding new associations "is transmitted to all nodes in the corresponding trunk." *Id.* ¶33. Thus, it would have been obvious to a POSITA that this must include at least the second line card. *Id.*; EX1003 ¶¶143–144.

f) Claim 3

The combination of Smith and Ishimori discloses claim 3. Smith discloses this limitation. §VII.A.2.j). To the extent this limitation is not disclosed by Smith, it is disclosed by the combination of Smith and Ishimori. Ishimori teaches the reception path information is associated with each other and learned, which is done by storing the record of association to the MAC table (*e.g.*, adding the record of the association to the FDB). EX1005 ¶2. The routes are transmitted via a "learn packet" that can "perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably." *Id.* ¶25. Ishimori discloses "a learn packet is transmitted to all nodes having the corresponding trunk," which

necessarily includes at least the second line card. *Id.* ¶33; EX1003 ¶¶145–147. In Ishimori, each of the communication cards, which includes the second line card, would add a record of the association to the FDB in response to the learn packet. EX1005 ¶25. A POSITA could use Ishimori's learn packet in place of Smith's MAC notification frames in order to send a message to the second line card to add an entry in its MAC table. EX1003 ¶147.

g) Claim 11[c]

The combination of Smith and Ishimori discloses claim 11[c]. §§VII.A.2.b)–A.2.d)d),VII.A.2.m),VII.B.2.a); EX1003 ¶148.

h) Claim 11[d]

The combination of Smith and Ishimori discloses claim 11[d]. §§VII.A.2.e)–i),VII.B.2.b)–d); EX1003 ¶149.

i) Claim 12: The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.

The combination of Smith and Ishimori discloses claim 12. §VII.B.2.e); EX1003 ¶150.

j) Claim 13

The combination of Smith and Ishimori discloses claim 12. §VII.B.2.f); EX1003 ¶151.

C. Ground 3: Claims 1, 4–7, 10–11, 14–17, and 20 would have been obvious over Smith in view of Ishimori in further view of Edsall.

Although Smith alone or in combination with Ishimori renders obvious claims 1–3, 6, 11–13, and 16, to the extent any elements are not rendered obvious by Smith and Ishimori, they would have been obvious in further view of Edsall. Petitioners incorporate by reference the analysis of Smith and Ishimori from above. §VII.B.2. Additionally, the combination of Smith, Ishimori, and Edsall renders obvious claims 4–5, 7, 10, 14–15, 17, and 20.

1. Overview of Ground 3

a) Edsall

Edsall is generally directed to techniques for updating and synchronizing "forwarding tables contained on line cards that are interconnected by a switch fabric of a distributed network switch." EX1006, Abstract. The "forwarding table" has an L2 portion that is "used to execute forwarding decision operations for frames forwarded among ports of the line cards." *Id.*, 5:66–6:4. Similar to Smith and Ishimori, "[i]f the frame is received at the ingress card for the first time," the ingress forwarding engine learns the source MAC address of this frame. *Id.*, 6:26–31. This involves "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.*, 6:31–34. Then, the ingress forwarding engine floods "copies of the fabric frame through its port

interfaces to all (egress) line cards of the network switch," called the "flood-to-fabric (FF)" operation. *Id.*, 6:34–39, 18:44–47.

A "novel MN [MAC notification] frame" complements the flooding operation. *Id.*, 6:46–50. The MN frame "involves use of a primary input (PI) indicator," which "denotes a primary input MAC address that is directly attached to a port of the line card associated with the forwarding table containing this entry." *Id.*, 6:50–56. As Edsall explains, the PI indicator is "asserted" for "a MAC address that is learned from a frame sourced through one of the ports of the line card, as opposed to being learned through the switch fabric." *Id.*, 6:56–60, 18:56–19:5. The frame additionally includes a POE (port-of-exit) field that "includes a plurality of bits, one for each port interface of the switch fabric." *Id.*, 6:24–25. The POE bit instructs the switch which port interfaces on which line cards should receive the MN frame. *Id.*, 9:47–50.

b) Motivation to Combine Smith and Ishimori with Edsall

A POSITA would have been motivated to combine Smith, Ishimori, and Edsall. EX1003 ¶¶89–93. All three references disclose methods of using MAC forwarding tables in a distributed network switch. EX1004 ¶54; EX1005 ¶9; EX1006, Abstract. It would have been obvious to a POSITA to modify the MAC forwarding tables of Smith to implement the various features found in Edsall's forwarding engine. *Id.* ¶89. Specifically, a POSITA could alter the MAC tables of

Smith to implement how Edsall's forwarding engine learns the source MAC address by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." EX1006, 18:39–44; EX1003 ¶89.

A POSITA could further implement Edsall's "flood-to-fabric (FF) operations" by modifying the MAC tables to include POE bits that determine which port interfaces on which line cards should receive the MN frames. *Id.*, 18:47–50. Finally, it would have been obvious to a POSITA to modify the MAC tables to include the PI indicator as taught in Edsall in order to keep track of whether a MAC address was learned from a local line card or a different line card. EX1003 ¶89. It would have also been obvious to a POSITA to further modify the MAC forwarding tables of Smith to implement the POE field in order to determine which ports should receive the MN frame. *Id.*

Additionally, Smith and Edsall are both assigned to the same assignee, and so a POSITA would have had a motivation to combine these references for this additional reason. *Abbot Vascular, Inc. v. Flexstent*, IPR No. 2019-882, Paper 48, 28–29 (Oct. 2, 2020); *Laird Techs., Inc. v. Garftech Int'l Holdings, Inc.*, IPR No. 2014-24, Paper 46, 30–31 (Mar. 25, 2015). Furthermore, creating entries in a MAC forwarding table and adding a PI indicator field and POE field to the MAC tables

would have been a modification a POSITA would have known how to make with a reasonable expectation of success. EX1003 ¶90.

Ishimori additionally teaches an aging process using a "hit bit." EX1005 ¶9. The "hit bit" keeps track of how long a MAC address has been in the MAC address table without having been refreshed. *Id.* If the MAC address has not been refreshed within a predefined time, then the MAC address is deleted from the buffer on the next cycle. *Id.* "As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." *Id.* ¶25. A POSITA would have understood that MAC addresses may become stale due to changes in the network and would have been motivated to look at solutions such as Ishimori's aging parameter for making sure the MAC table is up to date. EX1003 ¶92. Moreover, because the size of a MAC table is limited, a POSITA would have known older entries have to be aged to make space for newer entries. *Id.* A POSITA looking to implement this solution would have therefore been motivated to combine Smith and Ishimori and it would have been an easy application of Ishimori's methods to Smith's system. *Id.* For example, aging information could be added to the MAC address table described in Smith using known techniques of adding information to a table. *Id.*

Furthermore, one of skill in the art would have had a reasonable expectation of success in implementing this combination because it would have required a simple addition of Ishimori's aging process to Smith-Edsall. *Id.* ¶93. Smith, Ishimori, and

Edsall already leverage link aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within the knowledge of a POSITA to simply implement the additional aging process taught by Ishimori to Smith-Edsall, and a POSITA would have been able to do so with a reasonable expectation of success. *Id.*

2. Analysis of Ground 3

a) Claim 1[g]

To the extent this is not disclosed by the combination of Smith and Ishimori, Edsall discloses this. Edsall teaches the forwarding engine learns the *source* MAC address of a frame received at the ingress card for the first time. EX1006, 18:39–41. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.*, 18:42–44. "[I]f there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.*, 18:54–55. It would have been obvious to use Edsall's learning methods with the Smith-Ishimori MAC tables, and a POSITA would have had a reasonable expectation of success in doing so. EX1003 ¶¶153–154.

b) Claim 1[h]

To the extent the combination of Smith and Ishimori does not render obvious this limitation, it would have been obvious to combine Smith and Ishimori with Edsall. Edsall teaches the forwarding engine learns the *source* MAC address of a

frame received at the ingress card for the first time. EX1006, 18:39–41. Edsall further teaches "if there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.*, 18:54–55. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.*, 18:42–44. It then floods copies of the fabric frame to all the egress line cards of the network switch, which Edsall calls the "flood-to-fabric (FF) operation." *Id.*, 18:47–50. This "forces each forwarding engine associated with each egress card to either (i) update its current L2 forwarding table entry with the newly-learned source MAC address and index of the frame or, if there is not a current entry, (ii) learn the source address/index of the frame." *Id.*, 18:50–55.

It would have been obvious to a POSITA to add Edsall's learning method and flood-to-fabric operation to the Smith-Ishimori path learning operations. EX1003 ¶¶155–156. A POSITA would have been able to implement this teaching with a reasonable expectation of success, because the combination of Smith and Ishimori already teach a message of new associations are sent to the plurality of member line cards. EX1004 ¶63; EX1005 ¶16. It would have been within the knowledge of a POSITA to create new entries in a FDB when an association of a MAC address and port does not yet exist. EX1003 ¶156.

c) Claim 4: The method according to claim 3 and comprising marking the records in the respective

FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

The combination of Smith, Ishimori, and Edsall discloses claim 4. Edsall discloses the "PI indicator" (*e.g.*, marking the records) is asserted when a forwarding table entry for the MAC address is learned through one of the ports (*e.g.*, data packets transmitted via a port of the line card) "as opposed to through the switch fabric" (*e.g.*, received from another of the line cards). EX1006, 6:46–64. Additionally, during prosecution, the Examiner found Edsall disclosed this limitation, which the applicant did not dispute. EX1002, 82,120–122. Specifically, Edsall discloses the "PI indicator is asserted for a destination MAC address entry of the forwarding table on the egress card and the DI contained in the switched fabric frame (*i.e.*, the ingress DI) is ***different*** from the DI stored in this egress forwarding table (*i.e.*, the egress DI)." EX1006, 18:56–19:5; EX1002, 82. Thus, the PI indicator is different for the MAC address entry learned through one of the ports as opposed to through the switch fabric. EX1006, 6:46–64, 18:56–19:5; EX1003 ¶¶157–158.

d) Claim 5[a]: The method according to claim 4, and comprising: associating a respective aging time with each of the records;

The combination of Smith, Ishimori, and Edsall discloses claim 5[a]. Ishimori teaches each node has a hit bit that is set to "1" at the time of learning (*e.g.*,

associating a respective aging time with each of the records). EX1005 ¶9. The hit bit is then set to "0," so on the next aging process, "the corresponding MAC address is deleted from the buffer." *Id.* Thus, the hit bit is how Ishimori associates a respective aging time with each of the records. EX1003 ¶¶159–161. It would have been obvious to a POSITA to implement a "hit bit" on the Smith's lookup tables with Edsall's PI indicator in order to associate a respective aging time with each of the records. *Id.* ¶161.

e) Claim 5[b]: refreshing the records in the FDB responsively to further packets transmitted by the line cards; and

The combination of Smith, Ishimori, and Edsall discloses claim 5[b]. As described above, Ishimori teaches when the hit bit is "0," "the corresponding MAC address is deleted from the buffer." EX1005 ¶¶9, 11. Ishimori then teaches when the destination MAC address is received at card #0, the source MAC address is learned by "overwriting," and the hit bit is returned to "1" (*e.g.*, refreshing the records in the FDB responsively to further packets transmitted by the line cards). *Id.* ¶11. It would have been obvious to a POSITA implement Ishimori's "hit bit" on Smith's lookup tables with Edsall's PI indicator and to refresh the hit bit by learning the source MAC address of the data packets transmitted by the line cards. EX1003 ¶¶162–163.

f) Claim 5[c]: removing the records from the respective FDB if the records are not refreshed within the respective aging time.

The combination of Smith, Ishimori, and Edsall discloses claim 5[c]. Ishimori teaches if the hit bit remains "0" in the next aging process, then the corresponding MAC address is deleted (*e.g.*, removing the records from the FDB if the records are not refreshed within the respective aging time). EX1005 ¶¶9,12. It would have been obvious to a POSITA implement Ishimori's "hit bit" on Smith's lookup tables with Edsall's PI indicator and to remove the records from the respective lookup tables if the record has not been refreshed by the next aging process. EX1003 ¶¶164–165.

g) Claim 6

The combination of Smith, Ishimori, and Edsall discloses claim 6. Smith teaches sending a message that is a synchronization packet . §VII.A.2.k). Edsall teaches the "plurality of line cards" are "interconnected by a switch fabric 550," which could be the "switching core." EX1006, 8:20–27. A POSITA would have understood that the packet is sent from the first line card via Edsall's "switching fabric" to at least the second line card. EX1003 ¶166.

h) Claim 7: The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed

The combination of Smith, Ishimori, and Edsall discloses claim 7. EX1003 ¶¶167–169. Smith teaches its virtual network device includes several line cards, which include several interfaces. EX1004 ¶46. "[W]hen updating control protocol behavior of virtual link bundle 250(1), a user can simply access virtual network device sub-unit 122(1) (instead of accessing both virtual network device sub-units 122(1) and 122(2))." *Id.* ¶59. "Virtual network device sub-unit 122(1) can then automatically propagate to network device 122(2) any changes made by the user to the control protocols." *Id.* Smith teaches "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶62.

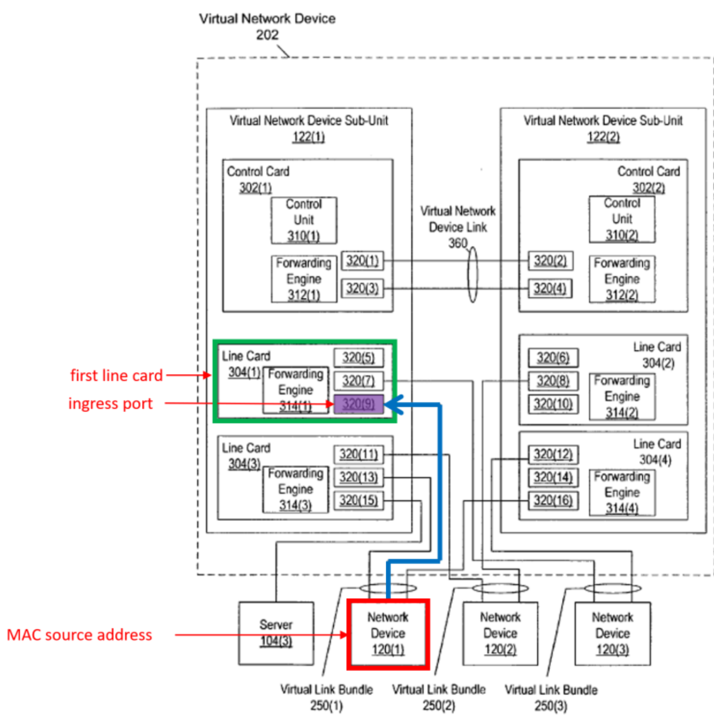


FIG. 3

Thus, if the record in the FDB of the first line card (*e.g.*, line card 304(1)) associates the MAC source address (*e.g.*, virtual device 120(1)) with a port different from the one of the ports on which the data packet was received (*e.g.*, a port other than 320(9)), the MAC notification frames will notify and update the L2 tables in the virtual network device 202, which would include at least the second line card (*e.g.*, line card 304(4)) to indicate the record has been changed. Similarly, Ishimori teaches the routes are transmitted via a "learn packet" (*e.g.*, synchronization packet). EX1005 ¶25. During the packet transfer, each of the communication cards learns information about the route (*e.g.*, the second line card). *Id.* ¶¶28,33; EX1003 ¶168.

Edsall further discloses the forwarding engine generates an MN frame (*e.g.*, synchronization packet) that may get sent to the SMC (switch management card) to ensure that FwdT0 (*e.g.*, the forwarding table) is synchronized. EX1006, 17:26–38. The forwarding engine also asserts an appropriate bit of the POE field (port-of-exit field) when generating the MN frame, which is a port different from one of the ports on which data was received. *Id.* It would have been obvious to a POSITA to modify the MN frame from Smith to include the POE field in order to note the port interface of the switch fabric. *Id.*, 6:24–25; EX1003 ¶169. This disclosure was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83,120–122.

i) Claim 10[a]: The method according to claim 1, and comprising: conveying a further data packet, received

from a further MAC source address, to the second line card for transmission over the network;

The combination of Smith, Ishimori, and Edsall discloses claim 10[a]. EX1003 ¶¶170–171. Smith teaches the uplink interface receives a data packet with the "sending device's MAC address" (*e.g.*, a MAC source address). EX1004 ¶54; *see also* §VII.A.2.e). Based on Figure 3, a further data packet received from a further MAC source address (*e.g.*, network device 120(3)) would be conveyed to the second line card (*e.g.*, line card 304(4)) for transmission over the network.

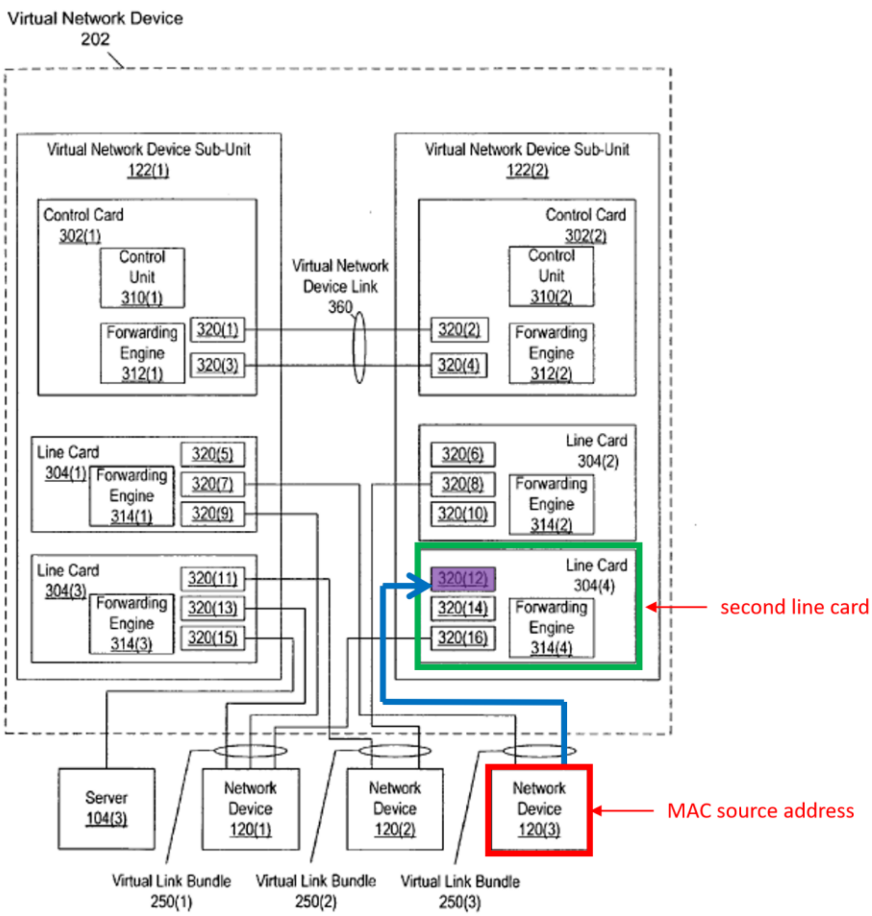


FIG. 3

To the extent Smith does not teach this claim, Edsall does. Edsall teaches in each subsequent frame, the encoded address recognition logic (EARL) circuit looks up the MAC address and "sends the corresponding rewrite information over the local bus after the frame" (*e.g.*, conveying a further data packet to the second line card for transmission over the network). EX1006, 14:22–34. This was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83–84,120–122.

j) Claim 10[b]: checking the further MAC source address against the records in the FDB of the second line card; and

The combination of Smith, Ishimori, and Edsall discloses claim 10[b]. EX1003 ¶¶172–173. Smith teaches the virtual network device sub-unit learns the source identifier of the sending device (*e.g.*, MAC source address). EX1004 ¶65; *see also* §VII.A.2.h). These identifiers are stored in a lookup table (*e.g.*, FDB) on virtual network device sub-unit. *Id.* ¶61. The MAC source address (*e.g.*, network device 120(3)) would be checked in the records of the FDB of said second line card (*e.g.*, line card 304(4)). EX1003 ¶172.

To the extent Smith does not disclose this limitation, Edsall does. Edsall teaches checking to see if the rewrite information matches (*e.g.*, checking the MAC source address against the records in the FDB of the second line card). EX1006, 14:22–34 ("The destination port circuitry (or, alternatively, a UDlink or central rewrite engine) matches the frame with the rewrite information and modifies the

frame as needed by replacing, *inter alia*, the destination and source MAC addresses."). This was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83–84,120–122.

k) Claim 10[c] responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.

The combination of Smith, Ishimori, and Edsall discloses claim 10[c]. EX1003 ¶¶174–175. Smith teaches the network device sub-unit 122(2) sends a MAC notification (*e.g.*, a message) to update the forwarding engines (*e.g.*, sending a further message to inform at least the first line card) when it learns of a new association (*e.g.*, the further record). EX1004 ¶63; *see also* §VII.A.2.h). "After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* A POSITA would understand that the MAC notification (*e.g.*, message) is therefore sent to inform at least the first member line card (*e.g.*, line card 304(1)) of the new association. EX1003 ¶¶174–175.

To the extent Smith does not disclose this limitation, Edsall does. Edsall teaches the forwarding engine "modifies the frame as needed by replacing" the "destination and source MAC addresses" (*e.g.*, adding a further record with respect to the MAC source address to the FDB). EX1006, 14:22–34. This was cited by the

examiner during prosecution and was not disputed by the applicant. EX1002, 83–84,120–122. It would have been obvious to a POSITA to use any of the learning methods from Smith, Ishimori, or Edsall, which all disclose updating a second line card with a further record and sending a message to inform at least the first line card of the further record. EX1003 ¶175.

l) Claim 11[a]

To the extent this is not disclosed by the combination of Smith and Ishimori, Edsall discloses this. The '400 patent describes the "switching core" as linking the multiple line cards. EX1001, 6:8–10. Edsall teaches the "plurality of line cards" are "interconnected by a switch fabric 550." EX1006, 8:20–27. Thus, a POSITA would have understood that the "switching fabric" in Edsall could be a "switching core." EX1003 ¶176.

m) Claim 11[d]

The combination of Smith, Ishimori, and Edsall renders obvious claim 11[d]. §§VII.A.2.p),VII.B.2.h),VII.C.2.a)–b); EX1003 ¶177.

n) Claim 14: The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

The combination of Smith, Ishimori, and Edsall discloses claim 14. §VII.C.2.c); EX1003 ¶178.

o) Claim 15: The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.

The combination of Smith, Ishimori, amd Edsall discloses claim 15. §§VII.C.2.d)–f); EX1003 ¶179.

p) Claim 16

The combination of Smith, Ishimori, and Edsall discloses claim 16. §VII.C.2.g); EX1003 ¶180.

q) Claim 17: The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.

The combination of Smith, Ishimori, and Edsall discloses claim 17. §VII.C.2.h); EX1003 ¶181.

r) Claim 20: The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the

network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

The combination of Smith, Ishimori, and Edsall discloses claim 20. §§VII.C.2.i)–k); EX1003 ¶182.

D. Ground 4: Claims 8–9 and 18–19 would have been obvious over Smith in view of Ishimori in further view of Zelig

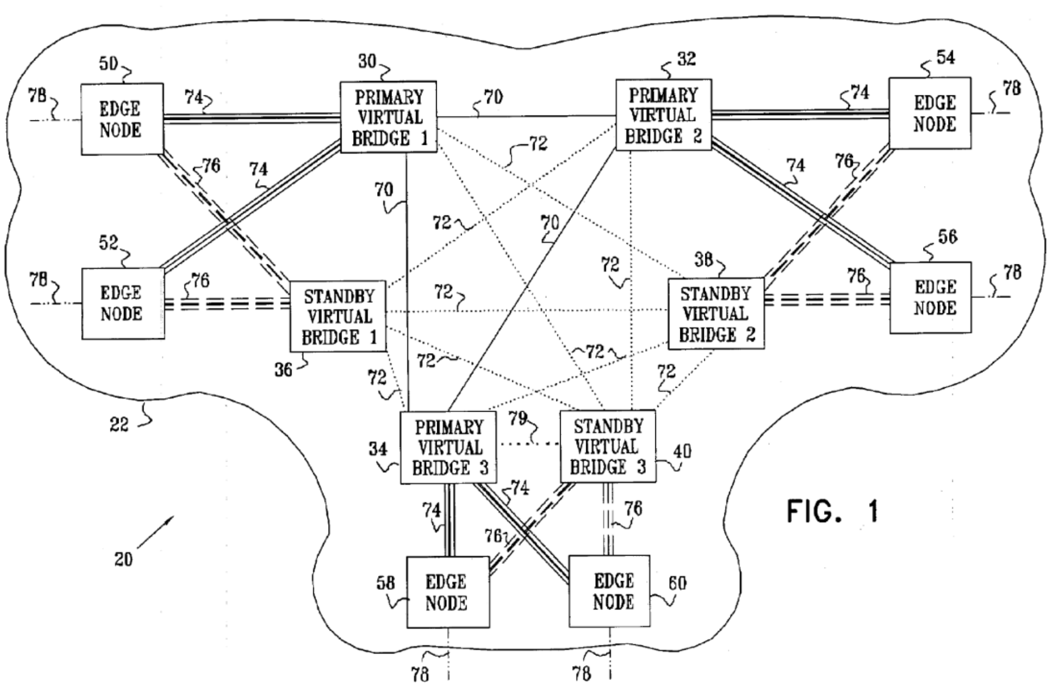
The combination of Smith, Ishimori, and Zelig renders obvious claims 8–9 and 18–19.

1. Overview of Ground 4

a) Zelig

Zelig is directed to a data communication network that "includes a plurality of primary virtual bridges, interconnected by primary virtual connections." EX1007, Abstract. Zelig discloses "MAC bridges that implement the 802.1D standard allow MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* ¶3. A MAC bridge "includes two or more MAC devices that interconnect the bridge ports to respective LANs." *Id.* The MAC bridges "maintain a database to map destination MAC address of the packets they receive to bridge ports." *Id.* ¶4. This database is built "by means of a learning

process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* The purported invention in Zelig is directed to providing "improved mechanisms for protection from failure in virtual private networks (VPNs)." *Id.* ¶18. Zelig thus discloses a data communication network including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" that are "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶21–24. "[E]ach of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table." *Id.* ¶27. The VPN 20 includes multiple primary virtual bridges. *Id.* ¶42.



b) Motivation to combine Smith and Ishimori with Zelig

A POSITA would have been motivated to combine Smith-Ishimori with Zelig. EX1003 ¶¶94–95. All three references disclose methods of using MAC bridges in a data communication network. EX1004 ¶54; EX1005 ¶9; EX1007, Abstract. Zelig teaches the MAC bridges "maintain a database to map destination MAC address of the packets they receive to bridge ports." EX1007 ¶4. This database is built "by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* In order to "improve[] mechanisms for protection from failure in virtual private networks (VPNs)," Zelig teaches a data communication network including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" can be "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶18,21–24.

A POSITA looking to improve against failure in VPNs would look to Zelig and implement Zelig's MAC bridges into the Smith-Ishimori system to create a separate MAC table for each MAC bridge as taught in Zelig. EX1003 ¶94. It would have also been obvious to a POSITA to configure each virtual MAC bridge to serve a respective VPN, as shown above in Figure 1 of Zelig. *Id.* ¶95. Furthermore, implementing the MAC bridges as taught in Zelig to the Smith-Ishimori data communication network would have been a modification that a POSITA would have known how to make with a reasonable expectation of success. *Id.*

2. Analysis of Ground 4

a) Claim 8: The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

The combination of Smith, Ishimori, and Zelig discloses claim 8. Zelig discloses its VPN 20 contains multiple primary virtual bridges 30, 32, and 34 (*e.g.*, multiple virtual MAC bridges in a Layer 2 VPN). EX1007 ¶¶42–43. It would have been obvious to a POSITA that the virtual bridges servicing VPN 20 are configured to serve that VPN. EX1003 ¶¶184–185. Zelig further teaches "each of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table . . . ." EX1007 ¶31. Thus, Zelig discloses each MAC bridge maintains its own MAC table, and because the MAC bridge serves only that VPN instance, the records associating the MAC addresses with the respective ports are maintained independently for each VPN instance. EX1003 ¶185.

b) Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

The combination of Smith, Ishimori, and Zelig discloses claim 9. Smith teaches the MAC notification (*e.g.*, message) is "distributed to any other forwarding engines within virtual network device sub-unit 122(2)." EX1004 ¶63. Zelig provides "VPN 20 is built around a virtual private LAN service (VPLS), operating within a network 22." EX1007 ¶42. It would have been obvious to a POSITA to adopt the notification scheme in Smith to the architecture in Zelig so the MAC notification (*e.g.*, message) is sent to all the line cards not just in the virtual network device sub-unit, but to all the line cards that serve the VPLS instances. EX1003 ¶186–187. Furthermore, the IEEE 802.1Q standards disclose a VLAN identifier (VID) as a twelve-bit field that "uniquely identif[ies] the VLAN to which the frame belongs." EX1008 §9.3.2.3. Thus, it would have been within the knowledge of a POSITA to identify the VPLS instance in the message, such as using the VID, in order to inform all the line cards in the VPLS. EX1003 ¶187.

c) Claim 18: The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

The combination of Smith, Ishimori, and Zelig discloses claim 18. §VII.D.2.a); EX1003 ¶188.

d) Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

The combination of Smith, Ishimori, and Zelig discloses claim 19. §VII.D.2.b); EX1003 ¶189.

E. Ground 5: Claims 9 and 19 would have been obvious over Smith in view of Ishimori, Zelig, and 802.1Q

The combination of Smith, Ishimori, Zelig, and 802.1Q renders obvious claims 9 and 19.

1. Overview of Ground 5

a) 802.1Q

The 802.1Q standard defines the architecture for Virtual Bridged LANs, its services, protocols, and algorithms. EX1008, Abstract. It was part of the effort of the IEEE to standardize virtual LAN services in bridged LANs. EX1003 ¶84. The standard discloses a twelve-bit VLAN identifier field is used to "uniquely identify the VLAN to which the frame belongs." EX1008 §9.3.2.3. Such a standard would have been known and within the knowledge of a POSITA. EX1003 ¶84.

b) Motivation to combine Smith, Ishimori, and Zelig with 802.1Q

A POSITA would have been motivated to combine Smith-Ishimori-Zelig with 802.1Q-1998. *Id.* ¶96. All four references pertain to MAC bridges in a data

communication network. EX1004 ¶54; EX1005 ¶9; EX1007, Abstract; EX1008, Abstract. Moreover, Zelig expressly cites to the 802.1Q standard. EX1007 ¶12. A POSITA would have therefore been motivated to look to the teachings of 802.1Q in combination with Smith-Ishimori-Zelig to modify the messages from Smith-Ishimori-Zelig with a VLAN identifier as disclosed in 802.1Q in order to identify the VPLS instance. EX1003 ¶96. Furthermore, modifying the Smith-Ishimori-Zelig message to add the VLAN identifier would have been a modification that a POSITA would have known how to make with a reasonable expectation of success. *Id.*

2. Analysis of Ground 5

a) Claim 9

The combination of Smith, Ishimori, Zelig, and 802.1Q discloses claim 9. §VII.D.2.b); EX1003 ¶191.

b) Claim 19

The combination of Smith, Ishimori, Zelig, and 802.1Q discloses claim 19. §VII.D.2.b),VII.D.2.d); EX1003 ¶192.

VIII. PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION

A. PTAB should not exercise its discretion to deny institution under *Fintiv*.

1. Factor 1: Institution will increase the likelihood of stay

The District Court has made clear a motion to stay prior to institution of an IPR would be premature, so Petitioners have not yet requested a stay. The Board

has repeatedly stated it will not speculate on how a district court will treat a motion to stay. *See, e.g.*, *VMware Inc. v. Intellectual Ventures II LLC*, IPR2020-00859, Paper No. 13, at 12 (PTAB Nov. 5, 2020) (informative). Thus, this factor is, at worst, neutral.

2. Factor 2: District Court schedule

In the Cisco litigation, January 3, 2023 is the deadline to submit a Joint Claim Construction Statement. EX1013. The *Markman* hearing is not until February 15, 2023. EX1014. No date has been set for trial. A Final Written Decision ("FWD") would be expected in the second quarter of 2024. The Dell litigation does have a scheduling order yet. The parties worked expeditiously to prepare and file this petition approximately four months after Cisco received Patent Owner's infringement contentions on August 3, 2022.

Furthermore, the current average time-to-trial for the W.D. Texas district court (for the Cisco litigation) is over two years. EX1023 (showing W.D. Texas median time-to-trial of 28.8 months); *In re Apple Inc.*, 979 F.3d 1332, 1350 (Fed. Cir. 2020). Petitioners note "a court's general ability to set a fast-paced schedule is not particularly relevant…where, like here, the forum itself has not historically resolved cases so quickly." *Id.* at 1344. The current average time-to-trial for the Delaware district court (for the Dell litigation) is much longer. EX1024 (median time-to-trial of 39 months). Accordingly, this factor weighs against discretionary

denial. *See Sand Revolution II, LLC v. Continental Intermodal Grp.*, IPR2020-01393, Paper 24 (June 16, 2020).

3. Factor 3: Petitioners' investment in IPR outweighs forced investment in litigation to date

The co-pending litigations are in their early stages, and the investment in them has been minimal. The *Markman* hearing is not for another couple months in the Cisco litigation and discovery has not begun. EX1014; *see PEAG LLC v. Varta Microbattery GMBH*, IPR2020-01214, Paper 8, 17 (Jan. 6, 2021). The Dell litigation does not yet have a scheduling order.

4. Factor 4: The Petition raises unique issues

Preliminary invalidity contentions in the Cisco litigation were served on Patent Owner on September 28, 2022. Instituting a proceeding will allow the Board to address the asserted prior art, and the issues will be narrowed in the co-pending litigations due to the estoppel provisions of 35 U.S.C. § 315(e)(2).

Moreover, there will be no overlap of prior art issues. If the Board institutes trial, Petitioners will cease asserting in the co-pending litigations the combination of references on which trial is instituted for the claims on which trial is instituted, to the extent Petitioners even assert the same combination in the litigations against them. This factor weighs against discretionary denial. *See Verizon v. Huawei*, IPR2020-01079, Paper 10, 38 (Jan. 14, 2021).

5. Factor 5: Whether the Petitioners and Defendants in the parallel litigation are the same party

Petitioners are defendants in co-pending litigations. While the *Fintiv* case indicates that a difference between the district court defendant and the petitioner may weigh against discretionary denial, nothing within the *Fintiv* case suggests the same party between the proceedings weighs in favor of discretionary denial. *See HP Inc. v. Slingshot Printing LLC*, IPR2020-01084, Paper 13, 9 (Jan. 14, 2021). Accordingly, this factor is neutral and should not be a basis for denying institution.

6. Factor 6: Other circumstances support institution

The prior art presented in this Petition renders the Challenged Claims unpatentable as obvious. The merits of Petitioners' arguments are compelling because they lead to a strong conclusion that one or more claims are unpatentable by a preponderance of the evidence. This factor therefore weighs against discretionary denial. As such, the *Fintiv* factors are either neutral or weigh against discretionary denial. Because this Petition was filed approximately four months before the statutory bar date, institution should not be denied on discretionary factors.

B. PTAB should not exercise its discretion to deny institution under *Becton* and *Advanced Bionics*

Furthermore, institution should not be denied under the framework set forth in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8

(Dec. 15, 2017) (§III.C.5, ¶1, precedential) and *Advanced Bionics LLC v. MED-EL Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6 (PTAB Feb. 13, 2020) (precedential). While Edsall was before the Examiner during prosecution, Edsall is only relied upon in this petition for disclosures of limitations that were not disputed by the applicant. Moreover, Smith alone and Smith in combination with Ishimori, which were not before the Examiner during prosecution form the basis of the disclosures for the independent claims here, to which Edsall serves as only a supplemental reference as disclosure for certain claim limitations. Therefore, the arguments presented in this Petition are not the same as, or substantially similar to, arguments previously presented to the Office. Thus, the first prong of the *Advanced Bionics* framework is not met, and there is no need to reach the second prong to resolve against discretionary denial under §325(d). *Oticon Med. AB v. Cochlear Ltd.*, IPR2019-00975 Paper 15, 20 (PTAB Oct. 16, 2019) (precedential).

C. Discretionary denial under *General Plastic* is not appropriate

The ’400 patent has not been challenged in any prior IPR petition, so none of *General Plastic* discretionary institution factors apply here. *See Gen. Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, 16 (Sept. 6, 2016) (Section II.B.4.i. precedential).

IX. CONCLUSION

Petitioners have established a reasonable likelihood that the Challenged Claims are unpatentable. Petitioners therefore respectfully request that IPR be instituted.

DATED: December 23, 2022

Respectfully Submitted,

/s/ Stuart Rosenberg

Stuart Rosenberg
(Reg. No. 60,772)
Gibson, Dunn & Crutcher LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: 650-849-5389
SRosenberg@gibsondunn.com

Attorney for Petitioners Dell Inc., Dell Technologies Inc., and Cisco Systems, Inc.

CERTIFICATION UNDER 37 C.F.R. § 42.24(d)

Under the provisions of 37 C.F.R. § 42.24(d), the undersigned attorney hereby certifies that the word count for Sections I and III-VIII of the foregoing Petition for *Inter Partes* Review is 13,969, according to the word count tool in Microsoft Word.

DATED: December 23, 2022

Respectfully Submitted,

/s/ Stuart Rosenberg

Stuart Rosenberg
(Reg. No. 60,772)
Gibson, Dunn & Crutcher LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: 650-849-5389
SRosenberg@gibsondunn.com

Attorney for Petitioners Dell Technologies Inc., Dell Inc., and Cisco Systems, Inc.

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a), (b) on the Patent Owner via US Express Mail and FedEx Priority Overnight of a copy of this Petition for *Inter Partes* Review and supporting materials at the correspondence address of record for the '400 patent shown in USPTO PAIR:

May Patents Ltd. (attorney/agent of record)
c/o Dorit Shem-Tov
P. O. Box 7230
Ramat-Gan, - 5217102
Israel

Corrigent Corporation (assignee of record)
291 Main St.
C/O New England Intellectual Property
West Newbury, Massachusetts 01985

DATED: December 23, 2022

Respectfully Submitted,

/s/ Stuart Rosenberg

Stuart Rosenberg (Reg. No. 60,772)
Gibson, Dunn & Crutcher LLP
1881 Page Mill Road
Palo Alto, CA 94304-1211
Tel: 650-849-5389
SRosenberg@gibsondunn.com

Attorney for Petitioners Dell Technologies Inc., Dell Inc., and Cisco Systems, Inc.

APPENDIX: CHALLENGED CLAIM LISTING

No.	Limitation
1[pre]	A method for communication, comprising:
1[a]	configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;
1[b]	configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards;
1[c]	providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;
1[d]	receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;
1[e]	conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;
1[f]	if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;
1[g]	checking said MAC source address of the data packet against records in said FDB of said first line card; and
1[h]	if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card of said plurality of member line cards.

No.	Limitation
2	The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
3	The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.
4	The method according to claim 3, and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
5[pre]	The method according to claim 4, and comprising
5[a]	associating a respective aging time with each of the records;
5[b]	refreshing the records in the FDB responsively to further packets transmitted by the line cards; and
5[c]	removing the records from the respective FDB if the records are not refreshed within the respective aging time.
6	The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via a switching core of the network node to at least the second line card
7	The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.

No.	Limitation
8	The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances
9	The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.
10[pre]	The method according to claim 1, and comprising:
10[a]	conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;
10[b]	checking the further MAC source address against the records in the FDB of the second line card; and
10[c]	responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.
11[pre]	A node for network communication, comprising:
11[a]	a switching core;
11[b]	a plurality of ports;
11[c]	a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards,

No.	Limitation
11[d]	wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.
12	The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
13	The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.
14	The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
15	The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.
16	The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

No.	Limitation
17	The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.
18	The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.
19	The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.
20	The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

EXHIBIT 3B

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

DELL TECHNOLOGIES INC.,
DELL INC.,
and
CISCO SYSTEMS, INC.,
Petitioners

v.

CORRIGENT CORPORATION,
Patent Owner.

Case No. IPR2023-00370

U.S. Patent No. 7,593,400

DECLARATION OF DR. NICHOLAS BAMBOS IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,593,400

TABLE OF CONTENTS

Page

TABLE OF APPENDICES

Document	Description
Appendix A	*Curriculum Vitae*
Appendix B	Text of Challenged Claims

TABLE OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 7,593,400 ("the '400 patent")
1002	Copy of Prosecution History of the '400 patent
1004	U.S. Patent Application Publication No. 2005/0198371 ("Smith")
1005	Certified English Translation of Japanese Patent Application No. 2005086668 ("Ishimori")
1006	U.S. Patent No. 6,735,198 ("Edsall")
1007	U.S. Patent Application Publication No. 2004/0133619 ("Zelig")
1008	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 1998 Edition ("802.1Q-1998")
1015	IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition
1016	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition
1017	IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition
1018	Kompella et al., *Virtual Private LAN Service*, IETF (December 2005)
1019	Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005)
1020	Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005)
1021	U.S. Patent No. 6,917,986
1022	U.S. Patent No. 7,974,223

I, Dr. Nicholas Bambos, hereby declare as follows:

I. INTRODUCTION

1. I have been retained on behalf of Dell Technologies Inc., Dell Inc., and Cisco Systems, Inc. (collectively, "Petitioners") as an independent expert in this Petition for *inter partes* review by the U.S. Patent and Trademark Office. As part of my engagement, I have been asked to provide analysis and expert opinions on the following topics: (1) the disclosure of the U.S. Patent No. 7,593,400 ("the '400 Patent"); and (b) the invalidity of claims 1–20.

2. In summary, it is my opinion that claims 1–20 of the '400 patent ("the Challenged Claims") are invalid in view of the prior art I discuss in this Declaration. In particular, it is my opinion that the prior art discussed herein discloses, teaches, or suggests the invention claimed in the Challenged Claims, rendering the Challenged Claims invalid as obvious over the prior art. The particular references and/or combinations that invalidate the Challenged Claims, as well as the reasons for my opinions and bases thereof, are set forth in detail below.

3. I am being compensated for my work on this case at my standard consulting rate. I am also being reimbursed for my expenses that I may incur. My compensation is in no way contingent upon the results of my study, the substance of my testimony, or the outcome of this case.

4. Based upon the facts and information available to me to date, I detail my opinions relevant to this petition for *inter partes* review. I am prepared to provide expert testimony on the opinions formed herein if asked about those issues by the Board or others.

5. Additionally, I may discuss my own work, teachings, and knowledge of the state of the art in the relevant time period. I may rely on handbooks, textbooks, technical literature, and the like to demonstrate the state of the art in the relevant period and evolution of the technology.

6. Throughout this declaration, I refer to specific line numbers or page numbers of the relevant prior art or other technical documents. The citations are intended to be exemplary as opposed to being exhaustive and are not intended to convey that the citations are the only source of evidence to support the propositions for which they are cited.

II. QUALIFICATIONS AND EXPERIENCE

7. I have knowledge of these technologies based upon education, training, or experience, of the matters set forth herein.

8. I am an R. Weiland Professor of Engineering at Stanford University, having a joint appointment in the Department of Electrical Engineering and the Department of Management Science & Engineering. I have also served as the

Fortinet Founders Department Chairman of the Management Science & Engineering Department from January 2016 to August 2020.

9. Before joining Stanford as an Associate Professor in 1996, I was since 1989 an Assistant Professor and later a tenured Associate Professor in the Electrical Engineering department of the University of California at Los Angeles (UCLA).

10. I received my Ph.D. from the University of California at Berkeley (1989) in Electrical Engineering and Computer Sciences (EECS). Also, from U.C. Berkeley, I received a M.S. in EECS (1987) and a M.A. in Mathematics (1989). I graduated in Electrical Engineering from the National Technical University of Athens-Greece (1984) with first class honors. While at U.C. Berkeley, I had been a U.C. Regents Fellow, a David Gale Fellow, an Earl Anthony Fellow, and an EECS Departmental Scholar.

11. At Stanford University, I head the Network Architecture and Performance Engineering research group, working on high-performance design of computer systems and networks. I have researched various high-performance design aspects of wireless and wireline networking (including switching) and computing technologies and have published numerous papers on such topics.

12. From 1999 to 2005, I was the Director of the Stanford Networking Research Center project. I have held the Cisco Systems Faculty Development Chair (1999-2003) in computer networking at Stanford University and have won the IBM

Faculty Development Award (2002) for research in performance engineering of computer systems and networks. I have also been the recipient of the National Young Investigator Award of the National Science Foundation (1992).

13. I have served as editor of various research journals (including the "Computer Networks" research journal), as technical reviewer for numerous networking and computing research journals, and on various technical panels for the National Science Foundation.

14. For over 30 years, I have done research in and taught networking and computing technology concepts and design principles (at Stanford since 1996 and at UCLA during 1989-96). I have graduated over 35 Ph.D. students who have later been in leadership positions in academia and the information technology industry (e.g., Stanford University, California Institute of Technology, Columbia University, New York University, University of Michigan; Cisco, IBM Labs, Qualcomm, ST Micro, Google, Intel, Nokia, MITRE, Sun Labs, Facebook, Twitter, etc.).

15. I have technical expertise in system architecture and high-performance engineering of computer networks/systems, have published over 200 peer-reviewed research papers in this field, have given numerous technical talks in this field world-wide, and have supervised numerous doctoral theses and research projects in this field at Stanford and UCLA. I am a named inventor on eight patents and have served as a technical expert witness in numerous patent litigation cases. My full

qualifications and experience are set forth in my Curriculum Vitae, which is attached as Exhibit A to this declaration.

III. PERSON OF ORDINARY SKILL IN THE ART

16. In evaluating the prior art references and other material, I have used the perspective of a person of ordinary skill in the art ("POSA") to which the patent is related as of the time of the patent's priority date. I am informed by Counsel that a POSA is presumed to be aware of pertinent prior art and the conventional wisdom in the art and is a person of ordinary creativity. I have applied this standard in my analysis throughout my declaration.

17. The '400 patent is entitled "MAC Address Learning in a Distributed Bridge" and is directed to the field of communication networks. '400 patent at 1:6–7. A POSA in the 2006 timeframe, which I am informed by Counsel is the earliest claimed effective filing date of the '400 patent, would have had at least 1) a bachelor's degree in electrical engineering, computer engineering/science, or a related field, and 2) either (a) a master's degree in electrical engineering, computer engineering/science, or a related field, or (b) two or more years of work or research experience in networking and computing; that is, more education can compensate for less professional experience.

18. I meet the criteria described above and I am a person with at least ordinary skill in the art pertaining to the '400 patent. I would have qualified as such

a person by at least 2006, which I understand is the earliest priority date of the '400 patent.

IV. APPLIED LEGAL PRINCIPLES

19. I am informed by Counsel for the Petitioners about the following legal standards, which I have applied throughout my analysis.[1]

A. Prior Art

20. I am informed by Counsel that a patent, published patent application, or other publication, must first qualify as prior art before it can be used to invalidate a patent claim.

21. I am further informed by Counsel that a U.S. or foreign patent qualifies as prior art to a patent claim if the date of application, issuance, and/or publication of the prior art patent is prior to the purported date of invention of the patent claim.

---

[1] I am informed by Counsel that the patent laws were amended by the America Invents Act (AIA), but that the earlier statutory requirements still apply to pre-AIA patents. I have been informed by Counsel that the '400 patent is a pre-AIA patent, so the pre-AIA requirements control. Unless otherwise stated, my understanding of the law about patent invalidity as set forth in this Declaration relates to the pre-AIA requirements.

22. I am additionally informed by Counsel that a printed publication, such as a technical publication, a magazine article, or newsgroup post, qualifies as prior art to a patent claim so long as the date of publication is prior to the purported date of invention of the patent claim.

23. I am moreover informed by Counsel that a U.S. patent qualifies as prior art to a patent claim if the application for the prior art patent was filed in the United States before the purported date of invention of the patent claim.

24. I am further informed by Counsel that, if a U.S. Patent incorporates another patent or document by reference, it is considered to be part of that same reference.

B. Anticipation

25. I am informed by Counsel that, once the claims of a patent have been construed, the second step in determining anticipation of a patent claim requires a comparison of the construed claim language to the prior art on a limitation-by-limitation basis.

26. I am further informed by Counsel that a prior art reference anticipates a claim, and thus renders the claim invalid, if all elements of the claimed process, system, or device are disclosed in the prior art reference, either explicitly or inherently.

27. I am also informed by Counsel that, under Section 102 of the Patent Act, claims of a patent are invalid for lack of novelty if they are anticipated by a single prior art reference. I am further informed by Counsel that a claimed invention is invalid as anticipated and hence lacks novelty if all of the limitations of the claim as construed by the Court are present in a prior art reference, or any document that the reference incorporates by reference. However, I am informed by Counsel that a limitation of the claim need not be shown directly in a reference so long as a POSA would have understood the limitation to be inherent, or necessarily present in the reference.

C. Obviousness

28. I am informed by Counsel that an invention may also be obvious, even though the invention is not identically disclosed or described in a single prior art reference, which is required for anticipation. This is true if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the relevant art.

29. I am further informed by Counsel the Supreme Court has urged caution in granting a patent based on the combination of elements found in the prior art. This is because a patent for a combination that only unites old elements with no change in their respective functions withdraws what already is known from the public,

giving it exclusively to the patentee. The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

30. I am also informed by Counsel that when a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, the invention is likely obvious. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill. Put another way, I am informed by Counsel that a court must ask whether the improvement is more than the predictable use of prior art elements according to their established functions. I am further informed by Counsel that this is a flexible standard and it is not confined to any particular formalistic conception.

31. I am also additionally informed by Counsel that, when conducting an obviousness inquiry, the analysis should consider the scope and content of the prior art, differences between the prior art and the claims of the challenged patent claims, and the level of ordinary skill in the art. Secondary considerations of nonobviousness can also be considered, which include commercial success enjoyed by devices practicing the patented invention, industry praise for the patented invention, copying

by others, and the existence of a long-felt but unsatisfied need for the invention. I am also informed by Counsel that there must be a nexus between these factors and the claims themselves for these secondary considerations to impact the analysis of obviousness.

32. I am informed by Counsel that there generally must be a motivation to combine the references in order to find obviousness. This motivation or reason to combine can arise from the references themselves, changing market dynamics, or from simple common sense.

33. I am further informed by Counsel that a single reference can render a patent claim obvious, even if that reference does not fully anticipate the claim. To determine whether there was an apparent reason to combine the known elements in the way a patent claims, it will often be necessary to look to interrelated teachings of multiple patents; to the effects of demands known to the design community or present in the marketplace; and to the background knowledge possessed by a person having ordinary skill in the art.

D. Claim Construction Standard

34. I am informed by Counsel that, during *inter partes* review, the same standard is used as during claim construction in district court civil actions. I am further informed by Counsel that, in the district court context and in this IPR, the words of a claim are generally given their ordinary and customary meaning. This is

the meaning the term would have to a person of ordinary skill in the art in question at the time of the invention, which generally means the time the application was filed.

35. I am further informed by Counsel that the specification informs the proper interpretation of a claim. This is because a person of ordinary skill in the art is deemed to have read the claim term not only in the context of the particular claim in which the dispute term appears, but in the context of the entire patent, including the specification. I am also informed by Counsel that the prosecution history and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art are also considered in ascertaining the meaning of claim terms.

V. TECHNOLOGY BACKGROUND

36. To put in context my opinions on the validity of the claims of the ’400 patent, in this section I provide some background on computer networks. All of the concepts discussed in this section were well-known and widely used well before the May 19, 2006 priority date for the ’400 patent.

A. Overview of Computer Networks

37. Computer communication networks are comprised of communication links connected to ports of communication nodes (*e.g.*, switches, routers, computer hosts) forming a graph of (communication) nodes and (communication link) edges.

Information communication between nodes occurs in data units called packets (or frames in some cases), which are finite sequences of 0/1 bits representing information in digital form. In general, each packet includes a destination address, that is, the identifier (ID) of the final/destination node that it wants to reach. Each intermediate node on the path of a packet (i.e., the sequence of links and nodes it traverses) from its source (i.e., the node that injected it into the network) to its destination receives the incoming packet via an input port from an upstream node and retransmits it via an output port to a downstream node, which typically depends on the destination address of the packet.

38. A rough analogy is that of viewing the packet as a "car," the communication network as a "freeway system," the communication nodes as "freeway intersections" and the communication links as the "freeway segments" connecting the intersections. When the "car" reaches a "freeway intersection" on a "freeway segment," the freeway signs direct (route) the car onto another "freeway segment" towards another "freeway intersection" and so on, until it reaches its destination and exits on a freeway ramp.

39. The path of a packet through a network, from its source node to its destination node, should have no loops under normal network operation. Clearly, a packet going around in loops and visiting any node more than once creates many problems, including 1) increasing the delay to deliver the packet to its destination

(since it has to lose time to traverse a loop of links and nodes to end up back at the node it entered the loop from) and 2) misusing the communication bandwidth of the nodes and links on the loop and, hence, causing unnecessary congestion on the network and delaying also other packets that are traveling though the network. Therefore, it is important that the link/node path traversed by each packet on the network be loop-free under normal network operation. The rough analogy of the communication network to the "freeway system" also makes it clear why we should avoid loops; a "car" should not loop around and visit the same "freeway segment" and/or "freeway intersection" twice at any point while traveling towards its destination under normal "freeway" conditions.

40. A tree is a graph where there is only one path (i.e., sequence of consecutive links/edges) to go from any node to any other one, as opposed to having more than one distinct path to go from some node to some other node. (i.e., a tree is a loop-free graph). Given any general graph (e.g., even with loops), a spanning tree of the graph is a tree (i.e., loop-free) subgraph that includes all nodes of the original graph, but possibly fewer links in order to break any loops that may exist in the original graph and reduce the original graph to a tree, hence, allowing no loops.

41. Spanning tree algorithms are used to construct spanning trees of communication network graphs in order to route packets on loop-free paths. Such algorithms are embedded in the Spanning Tree Protocol (STP) which runs on

computer communication networks in order to detect and remove loops, so that packets can be forwarded to their destinations on loop-free paths.

B. Switches, Learning MAC Tables, Spanning Trees

42. The Open Systems Interconnection (OSI) reference model is a layered modular abstraction of computer communication networks, especially with respect to network protocols (*i.e.*, operational schemes or mechanisms).

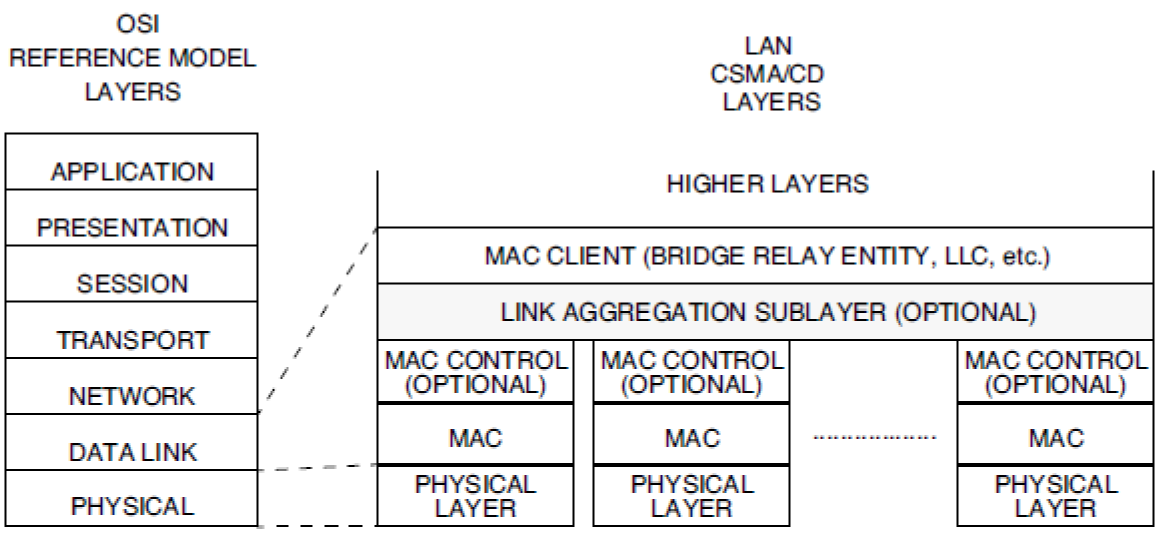


Figure 43–1—Architectural positioning of Link Aggregation sublayer

(Ex. 1017, Section 3, Fig. 43-1).

43. Layer 1 (L1) is the Physical layer, layer 2 (L2) is the Data Link layer and layer 3 (L3) is the Network layer; on top of those layers others are also defined. A sub-layer of the Data Link layer L2 is the Media Access Control (MAC) one. For example, a widely used computer communication (family of) protocol(s) generally

conforming to the L1, L2, L3 layers of the OSI model is the Ethernet (family of) protocol(s), which has been standardized by IEEE under the name 802.3.

44. Each network port/interface device on a network node has a unique hardware identifier (ID), which is called "MAC address" and is generally inserted into the device by its manufacturer. MAC addresses are used by L2 switches (also called bridges historically) to switch packets at network nodes from input ports to output ports. L2/MAC switches/bridges can interconnect separate physical LANs into a unified logical network.

45. L2 switching nodes, including in the context of Ethernet IEEE 802.3 technology, operate at the L2/MAC layer to allow transfer of data packets called "frames" from switch input ports to output ports. Each frame includes the MAC address of its source node and the MAC address of its destination node. Each node maintains a MAC table, which specifies at which output port the switch should output an incoming/received frame, based on its MAC destination address. That is, when the switch receives a frame, it "looks" at the frame's MAC destination address and "consults" its MAC table to find the port from where it should next transmit the frame to continue on the path to its destination. Thus, each switch's MAC table maps the incoming frames' MAC destination addresses to the switch's output ports, in order to transmit each received frame on to the downstream node on its path to its destination.

46. The MAC tables are "learned" dynamically and distributedly. Indeed, when an incoming frame is received at a port of the switch, the latter "looks" at the source MAC address of the frame, and infers and "learns" that this MAC address is reachable through that port, and "keeps" that information in its MAC table. Thus, the MAC table is (backward) "learned" by the switch, in the sense that if the switch sees an incoming frame with a certain source MAC address at one of its ports, then it infers and "learns" that this MAC address is reachable from that port (as a destination MAC address now). Therefore, next time the switch receives a frame with that MAC address as a destination MAC address, it will output the frame on that port.

47. When a switch is first connected to the network in a "plug-and-play" manner, its MAC table is empty and the switch "knows nothing" – it has to autonomously "learn as it goes." As mentioned before, when it receives an incoming frame at a port it "learns" that the source MAC address of the frame is also reachable as a destination MAC address from that port and "records" this information in its MAC table (if it hasn't done so already with a previously received such frame). When the switch receives a frame at one of its ports, the switch looks at the destination MAC address of the frame and consults its MAC table: 1) If it finds an entry that this destination MAC address is reachable from one of its ports, then it transmits the frame on that port. 2) If it does not find such an entry, the switch

transmits the frame on all its ports (floods it to its ports), except on the one where it received the frame from. As the switch operates, its MAC table is populated by more and more information entries mapping destination MAC addresses to switch ports from where these destination MAC addresses are reachable. After a while the MAC table of the switch has been built up enough to give it visibility into the network to forward frames to their destinations efficiently.

48. Of course, in general there may be multiple redundant links and paths to get from a source to a destination port on a network of switches for at least purposes of increasing reliability and robustness to component failures. Such a topology may include path loops which can cause frames to go in circles. Besides the detrimental effects mentioned above, allowing frames to go in loops is additionally detrimental to network "stability" in various ways.

49. For example, suppose that switch A is connected to switch B via two links, the first one connecting port A1 of A to port B1 of B, and the second connecting port A2 of A to port B2 of B. When a frame with a previously unobserved (by A and B) destination MAC address arrives to switch A on a port A3, it is output on both ports A1 and A2 to B1 and B2 respectively. Since the destination MAC address is also previously unobserved by switch B, when the frame is received by B1 it will be output on B2 (to A2) and also the (same) frame received on B2 will be output on B1 (to A1), and so on and so forth, which could cause an infinite

retransmission loop of the frame. To prevent formation of path loops, the switch network runs a Spanning Tree Protocol, which identifies redundant links and blocks transmission on them, leaving a single path to go from any source to any destination on the switch network.

50. Component failures, like port/link failures or whole switch/node failures, or even planned connection of new components to the network, or additionally movement (disconnection and reconnection) of existing components to other parts of the network cause reconfiguration of the whole network. Therefore, 1) the Spanning Tree Protocol will have to be rerun on the switches to identify a new spanning tree to forward frames from their sources to their destinations and 2) at least the MAC table entries corresponding to the topology changes will have to be purged or flushed and then "relearned" on the switches, as the old ones do not map to the new network topology any more.

C. Link Bundling and Link Aggregation

51. Multiple links (and their corresponding ports) connecting switches can be "bundled together," for example, into a Link Aggregation Group in some cases for various reasons, including increased bandwidth, traffic load balancing, scalability, increased reliability, etc. These Link Aggregation Groups are treated as a single logical link. For example, the IEEE 802.3 standard (as shown in Fig. 43-1 of Ex. 1017, Section 3, reproduced above) shows a Link Aggregation Sublayer of Layer

2 (*i.e.*, the Data Link layer of the OSI model), which can combine a number of individual physical links into a single logical link and present a single MAC interface to the MAC client.

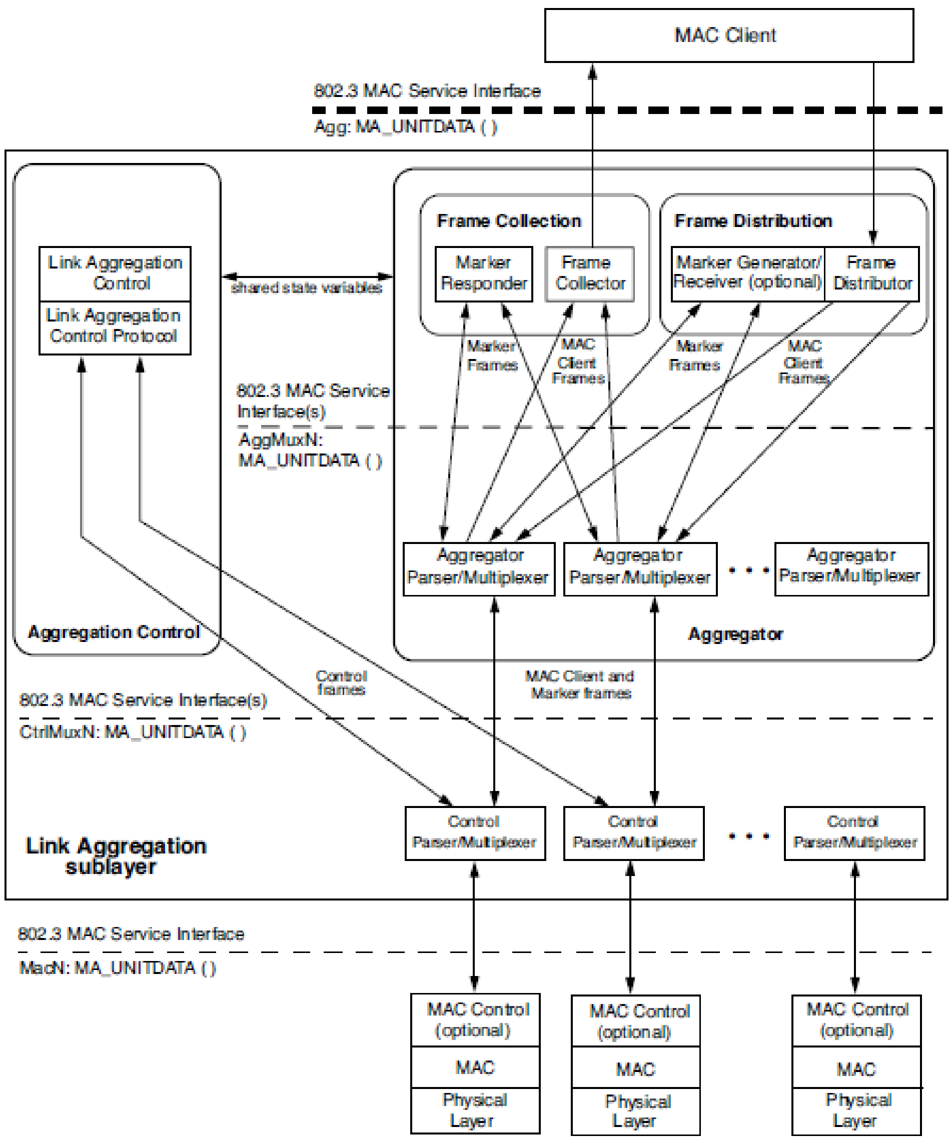


Figure 43–2—Link Aggregation sublayer block diagram

52. The architecture of this sublayer is shown in Fig. 43-2 of Ex. 1017 (IEEE 802.3 Part 3: Carrier sense multiple access with collision detection

(CSMA/CD) access method and physical layer specifications), Section 3 reproduced above showing the Frame Collection/Distribution modules and the aggregation control module, including the Link Aggregation Control Protocol (LACP) module.

53. "Link bundles" or LAGs across multiple physical switches would provide redundancy by eliminating a single point of failure: if one of the link/switches in the "bundle" or LAG went down, the other links/switches could still operate.

VI. BACKGROUND OF THE '400 PATENT

A. Summary of the '400 Patent

54. The '400 patent relates to "communication networks, and … distributed bridging systems." Ex. 1001 at 1:6–9. "Local Area Networks (LANs) connect computing systems together at the Layer 2 level. The term "Layer 2" refers to the second layer in the protocol stack defined by the well-known Open Systems Interface (OSI) model, also known as the logical link, data link, or Media Access Control (MAC) layer." *Id.* at 1:13–17. The '400 patent incorporates by reference the IEEE 802.1D standard and describes that "MAC bridges that implement the 802.1D standard allow MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* at 1:27–30. The specification also describes that "Recently, various means have been proposed and

developed for transporting Layer-2 packets, such as Ethernet frames, over high-speed, high-performance Layer-3 packet networks." *Id.* at 1:47–51.

55. At the time of the invention, "a number of authors have described methods for creating a virtual private LAN service (VPLS), which links different LANs together over an IP network." *Id.* at 2:9–12. Further, the specification describes that "Link aggregation (LAG) is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link (referred to as the "LAG group")." *Id.* at 2:37–40. "Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic." *Id.* at 2:40–42. The '400 patent states that "For ethernet networks, link aggregation is defined by Clause 43 of IEEE Standard 802.3, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (2002 Edition), which is incorporated herein by reference." *Id.* at 2:43–47.

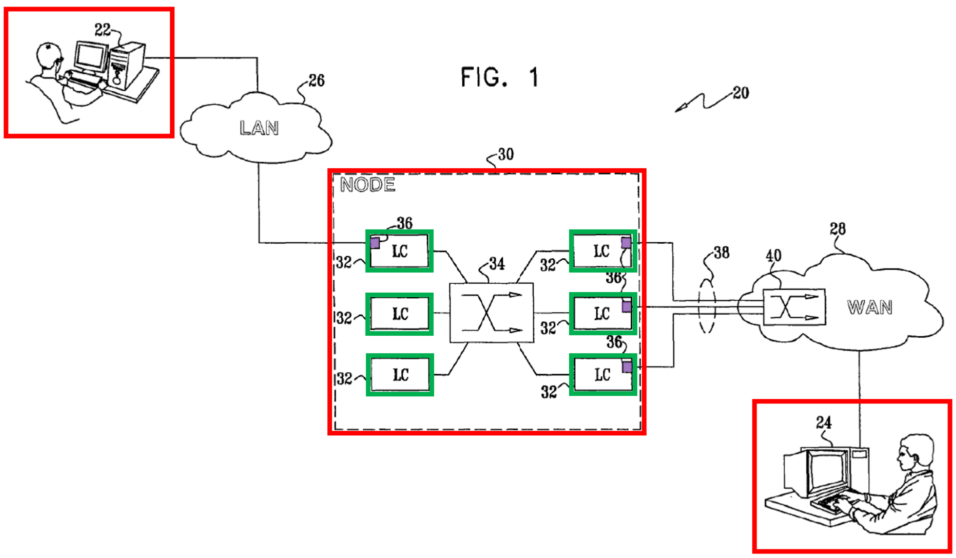
56. Figure 1 of the '400 patent (reproduced below) shows that "terminal **22** is connected to a LAN, such as an Ethernet LAN, while terminal **24** is connected to a wide area network (WAN) **28,** such as the Internet or another Layer 3 network. The VPLS, however, permits the users of terminals **22** and **24** to communicate with one another as though they were connected to the same LAN domain." *Id.* at 5:57–62. Further, the '400 patent states, "The specific configuration of LAN **26** and WAN **28**

is shown in FIG. **1** purely by way of illustration, and the principles of the present invention may be applied in substantially any network configuration that supports the provisioning of Layer 2 virtual private networks." *Id.* at 6:2–7. The '400 patent further describes Figure 1 as follows:

> In the exemplary configuration shown in FIG. 1, a network node 30 links LAN 26 and WAN 28. Node 30 comprises multiple line cards 32, linked by a switching core 34. Line cards 32 have ports 36, which connect to other nodes in LAN 26 and WAN 28 (and possibly in other networks, as well). Typically, each line card comprises multiple ports, although only a few ports are shown in FIG. 1. In the description that follows, ports 36 are assumed to be Ethernet ports, for the sake of simplicity of explanation. Alternatively, some or all of the line cards may comprise ports of other types, and may connect to other types of networks, such as Internet Protocol (IP) networks. For example, in an alternative embodiment (not shown in the figures), WAN 28 comprises a Resilient Packet Ring (RPR) network, and some of line cards 32 thus comprise RPR interfaces. Features of a network node that may be used to connect an Ethernet network to a RPR network are described, for example, in U.S. patent application Ser. No. 10/993,882, filed Nov. 19, 2004, which is assigned to the assignee of the present patent application

> and whose disclosure is incorporated herein by reference. Additionally or alternatively, line cards 32 may connect to tunnels, such as Multi-Protocol Label-Switching (MPLS) tunnels, through WAN 28 via appropriate label-switched routers in the WAN.

Id. at 6:8–30.



57. As stated above, a network node 30 "comprises multiple line cards 32, linked by a switching core 34," and each line card has ports 36, which connect to other nodes in LAN 26 and WAN 28. *Id.* at 6:8–12. "MAC bridges maintain a forwarding database (FDB) to map destination MAC addresses of the packets they receive to bridge ports. The bridge builds the forwarding database by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received. When the bridge receives an incoming packet whose destination address is not found in the database, it floods

(*i.e.*, broadcasts) the packet through all its available ports, except the one through which the packet arrived. Other MAC bridges that do not recognize the destination address will further flood the packet to all the relevant ports. Through the flooding mechanism, the packet will eventually traverse all interconnected bridges at least once, and will ultimately reach its destination." *Id.* at 1:33–46. The '400 patent states:

> Each of the line cards may typically serve as both ingress and egress for data packets and has a respective MAC forwarding database (FDB) that is shared by the ingress and egress functions. When an ingress line card receives an incoming data packet over the VPN on one of its ports, it consults the FDB in order to choose the line card and port through which the packet should be forwarded based on the MAC destination address (or floods the packet through the ports in the VPN when the MAC destination address does not appear in the FDB).

Id. at 3:7–17. As I discussed above, this method of flooding was well known at the time of the '400 patent. The '400 patent allegedly identified a need for better MAC address learning at the bridges within a VLAN to mitigate the need to flood (or broadcast) messages when the destination MAC address was unknown, as follows:

> When the forwarding destination of a packet is a link aggregation group (LAG), LAG member selection (i.e., selection of the link over which

> the packet is to be forwarded) is typically performed on the ingress line card. In the absence of the synchronization method described above, other members in the LAG may not receive such packets for transmission, so that the FDB of the corresponding line cards will not be updated. When these line cards receive incoming packets, the result may be constant flooding, since the FDB is incomplete. The synchronization mechanism described herein avoids this problem by **updating the FDB in all line cards in the LAG group (or across the entire VPN instance**) within the node. Typically, when the transmitting line card transmits the data packet via a port that belongs to a LAG group, the synchronization message sent by the line card identifies the VPN instance and the incoming port. The **other line cards in the same LAG group (as well as all the other line cards serving this VPN instance**) can use this information to learn the MAC address association **even when these other line cards have not yet received packets from the MAC address in question**.

Id. at 3:34–53 (emphasis added).

58. Referring to Figure 2, the '400 patent describes:

> Upon receiving an incoming packet from switch **40,** port **36** passes the packet to ingress path **54.** Packet processor **52** identifies the VPLS

(typically by a lookup and classification process based on certain packet header fields), extracts the other key parameters from the incoming packet (including the MAC destination address (DA), and optionally, the VLAN 30 identifier), and uses the key to query database **58.** If the record is found, the packet processor adds a tag to the packet indicating the egress port through which the packet should be forwarded, as well as the ingress port through which the packet was received. If the output interface indicated by the record is a LAG group, the packet processor selects one of the physical ports in the LAG group (using a hash function, for example), and tags the packet for transmission via the selected port. The packet processor then passes the tagged packet to switching core **34,** which conveys the packet to egress path **56** of the appropriate port.

When packet processor **52** receives a packet on ingress path **54** for whose key there is no a corresponding record in database **58,** however, it tags the packet for flooding. In this case, switching core **34** will pass the packet for transmission via all the ports (other than the ingress port through which the packet was received) that are used by this VPLS instance. For each LAG group serving the VPLS

instance, however, the flooded packet is transmitted via only one port in the group.

Id. at 7:25–49. That is, upon receiving an incoming packet, "[i]f the record is found, the packet processor adds a tag to the packet indicating the egress port through which the packet should be forwarded." *Id.* at 7:31–34. If there is "no . . . corresponding record in the database . . . however, it tags the packet for flooding." *Id.* at 7:42–44. While the "switching core 34 will pass the packet for transmission via all ports . . . that are used by this VPLS instance," "[f]or each LAG group serving the VPLS instance, however, the flooded packet is transmitted via only one port in the group." *Id.* at 7:47–49.

59. Synchronization messages ("SYNC") are sent at regular intervals "to report each SELF entry that it has created in the FDB 58 to the other line cards 32 in node 30." *Id.* at 8:17–22. The MAC table differentiates between SELF entries, which are entries that are learned by the packet processor on the line card and SYNC entries. *Id.* The '400 patent uses an aging mechanism that is applied to the "MAC database 58 in order to remove records that are no longer in effect and free space for new records." *Id.* at 9:4–6. A record is removed from the database "if a predetermined aging time elapses following the timestamp without a further packet having been received with the same key." *Id.* at 9:9–11. If, on the other hand, the current packet matches a record in the FDB, "the packet processor refreshes the

timestamp of the record," and "forwards the packet to the appropriate output port." *Id.* at 9:29–31.

B. File History of the '400 Patent

60. As I state below, in forming my opinions, I have also reviewed the file history of the '400 patent.

61. The '400 patent's application was filed on May 19, 2006. Ex. 1001, [22]. The Examiner rejected certain pending claims as anticipated by Edsall, or obvious over Edsall in view of another reference. Ex. 1002, 81–85. During an interview between the applicants and the Examiner, it was discussed that if the limitation "link aggregation group having a plurality of ports" in claim 1 were amended to recite the claim term as defined by the specification (at 2:37–40), then the amended claim would overcome Edsall. *Id.* at 112. The applicant therefore amended the claim limitation to add "a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link." *Id.* at 114. The applicant made no other amendments to overcome the rejections over Edsall. The Examiner allowed the claims, and the patent issued on September 22, 2009. *Id.* at 128, 160.

62. Based on my review of the record, I understand that the claims were amended to add "a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link"

to overcome the rejection by Edsall, which corresponds to the language found in claims 1[b] and 11[c]. My opinions do not rely on Edsall for disclosure of this particular claim language that was added to claims 1[b] and 11[c]. Instead, it is my opinion that claim 1[b] and 11[c] are disclosed by Smith, which I describe in more detail below.

VII. MATERIALS AND PRIOR ART CONSIDERED

63. I have considered information from various sources in forming my opinions. I have drawn on my decades of experience in this field. I have employed methods and analyses of a type reasonably relied upon by experts in my field in forming opinions or inferences on the subject. Additionally, in preparing this Declaration, I have relied upon ’400 patent and its file history, the exhibits cited to below, and the additional exhibits listed at the beginning of this Declaration.

64. In this Declaration, I provide five prior art references with exemplary citations identifying the relevant features related to the claim language of the Challenged Claims.[2] I have also described combinations of these prior art references that, when combined, render the Challenged Claims of the ’400 patent obvious. It is my opinion that the Challenged Claims of the ’400 patent are rendered obvious in light of the prior art specifically discussed in this Declaration.

[2] The complete text of the Challenged Claims is set forth in Appendix B hereto.

65. The citations presented in the body of this declaration correspond to the teachings disclosed in exemplary items of prior art that identify: (1) the problem(s) confronted by one of ordinary skill in the art; (2) an express suggestion to make one or more of the combinations; (3) an implicit suggestion to make one or more of the combinations; or (4) the knowledge of those skilled in the art as to the applicable field of technology at the relevant timeframe.

66. It is my opinion that a person of ordinary skill would consider at least the prior art references below:

Prior Art Reference	Date of Publication	Exhibit No.
U.S. Patent App. Pub. No. 2005/0198371 ("Smith")	September 8, 2005	Ex. 1004
Japanese Patent App. No. 2005086668 ("Ishimori")	March 31, 2005	Ex. 1005
U.S. Patent No. 6,735,198 ("Edsall")	May 11, 2004	Ex. 1006
U.S. Patent App. Pub. No. 2004/0133619 ("Zelig")	July 8, 2004	Ex. 1007
IEEE 802.1Q-1998 ("802.1Q-1998")	March 8, 1999	Ex. 1008

67. Specifically, it is my opinion that the claims of the '400 patent would have been obvious as follows:

’400 Patent Claims	Basis of Invalidity
1, 3, 6, 11, 13, 16	Obvious over Smith
1–3, 11–13	Obvious over Smith in view of Ishimori
1, 4–7, 10, 11, 14–17, 20	Obvious over Smith in view of Ishimori in further view of Edsall
8–9, 18–19	Obvious over Smith in view of Ishimori in further view of Zelig
9, 19	Obvious over Smith in view of Ishimori in further view of Zelig and 802.1Q-1998

68. I am also not aware of any secondary considerations of non-obviousness that affect my conclusions regarding the obviousness of these claims.

69. Other references and knowledge described earlier in this declaration are relevant to show the state of the art (*i.e.*, what a person of ordinary skill would have known when using the references identified above) or to use in combination with the references above to render the Challenged Claims invalid. I have considered also the following, which are also either cited or incorporated by reference in the ’400 patent:

a. IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition (Ex. 1015)

b. IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition (Ex. 1016)

c. IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition (Ex. 1017)

d. Kompella et al., *Virtual Private LAN Service*, IETF (December 2005) (Ex. 1018)

e. Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005) (Ex. 1019)

f. Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005) (Ex. 1020)

g. U.S. Patent No. 6,917,986 (Ex. 1021)

h. U.S. Patent No. 7,974,223 (Ex. 1022)

VIII. INVALIDITY ANALYSIS OF THE CHALLENGED CLAIMS

70. Below is a detailed analysis of the bases of invalidity identified above.

71. I note that claim 1 in the ’400 patent is a method claim and claims 2 through 10 depend from claim 1; claim 11 is an apparatus claim and claims 12 through 20 depend from claim 11. I am informed by Counsel that a method claim is a claim that describes a series of acts or steps for performing a desired function or accomplishing an intended result. I am informed by Counsel that an apparatus claim is a claim that defines the invention by its physical or structural components and their relationships. Because of the two claim types, there exist slight differences in the claim language (*e.g.*, “sending” versus “send”). Additionally, the apparatus

claim 11 recites a switching core, which is not found in independent claim 1. However, besides these differences, the differences found in the other claim limitations are immaterial for the purposes of my conclusions regarding unpatentability and do not affect my analysis. For my analysis, I have grouped claim limitations where they are substantively similar or where it made sense to discuss them together.

A. Overview of Prior Art References

1. Smith

72. Smith is directed to a virtual network device comprising interface bundles, which are managed as a single logical interface. Smith discloses:

> A virtual network device includes several different virtual network device sub-units, which collectively operate as a single logical network device. An interface bundle includes interfaces in more than one of the different virtual network device sub-units included in the virtual network device. The interface bundle is coupled to a virtual link bundle, which connects the virtual network device to another device. The interface bundle is managed as a single logical interface.

Ex. 1004, Abstract; *see also id.* ¶ 34. Smith teaches that "[t]he communication links are configured to be managed as a single link. When the first network device sends a packet to the virtual network device via the virtual link bundle, the first network

device selects one of the communication links on which to send the packet. Each packet sent between the virtual network device and the first network device is sent via only a one of the communication links." *Id.* ¶ 9.

73. As depicted in Figure 3, virtual network device 202 is coupled to other network devices 120(1)–120(3). *Id.* ¶ 44. The virtual network device consists of virtual network device sub-units 122(1) and 122(2), which includes several line cards 304(1)–304(4). *Id.* ¶ 46. "In virtual network device sub-unit 122(1), line card 304(1) includes forwarding engine 314(1) and interfaces 320(5), 320(7), and 320(9). Interface 320(7) is coupled to network device 120(3). Interface 320(9) is also coupled to network device 120(1)." *Id.* ¶¶ 46–47. "Line card 304(4) includes forwarding engine 314(4) and interfaces 320(12), 320(14), and 320(16). Interfaces 320(12) and 320(16) are respectively coupled to satellite network devices 120(3) and 120(1)." *Id.* ¶ 48. "Interfaces 320(13), 320(9), and 320(16), which are each coupled to network device 120(1) by virtual link bundle 250(1), form an interface bundle (e.g., an EtherChannel (TM) port bundle)." *Id.* ¶ 53. "Similarly, interfaces 320(11) and 320(8) form another interface bundle that is coupled to network device 120(2) by virtual link bundle 250(2). Interfaces 320(7) and 320(12) form a third interface bundle that is coupled to network device 120(3) by virtual link bundle 250(3)." *Id.*

74. The interface bundle is described by Smith as follows:

> One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using **a link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**. For example, an EtherChannel (TM) port bundle can be formed from several ports on a switch, each of which is coupled to a respective link in a group of links coupling that switch to another switch. Once an EtherChannel (TM) port bundle is formed, the port bundle can be managed as a single bridge port by routing protocols such as spanning tree, thus simplifying management of the redundant links.

Id. ¶ 6 (emphasis added).

> Various embodiments of methods and systems for implementing interface bundles in virtual network devices are disclosed. A virtual network device includes several different virtual network device sub-units, which collectively operate as a single logical network device. An interface bundle includes interfaces in more than one of the different virtual network device sub-units included in the virtual network device.

Id. ¶ 8.

75. Additionally, Smith teaches that:

> The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be operated as a **single logical link, referred to herein as a virtual link bundle**. Network device 120(1) operates the two links coupling network device 120(1) to virtual network device 202 as a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corresponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, **virtual link bundles** 250(1) and 250(2) are each operated as an EtherChannel (TM) or **as an aggregated link (as described in IEEE 802.3)**.

Id. ¶ 36 (emphasis added).

76. When a packet is received at an uplink interface, the virtual network device sub-unit can "learn that the sending device's MAC (Media Access Control) address is "behind" uplink interface 320(13) by associating the MAC address with the logical identifier of uplink interface 320(13)." *Id.* ¶ 54. The sub-unit then informs each forwarding engine within the virtual device of this association (between the MAC address and the logical identifier of the uplink interface). *Id.* As

Smith teaches, "packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier." *Id.*

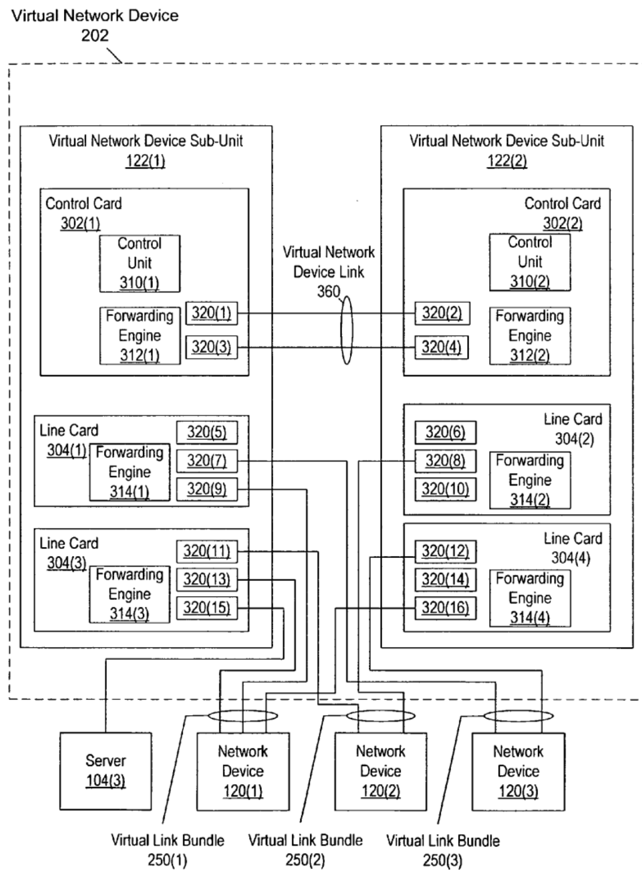


FIG. 3

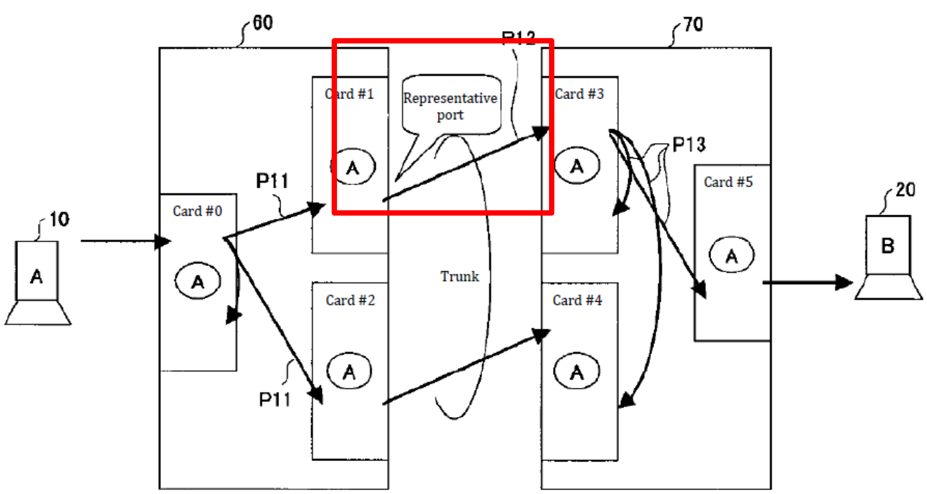
77. In order to keep its MAC tables updated, Smith uses "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶ 62. If a "forwarding table already includes an entry associating the destination address with a port of one of the network devices," that forwarding engine will generate "a MAC notification identifying this association, which is distributed to any other forwarding engines within virtual

network device sub-unit 122(2)." *Id.* ¶ 63. "If there is no hit in the forwarding table, as determined at 407, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at 409." *Id.* ¶ 66. But to take advantage of the virtual link bundles, Smith further specifies that as a result of the virtual link bundling, the data packet is "sent via only [] one of the communication links." *Id.* ¶ 9.

2. Ishimori

78. Ishimori is directed to a packet forwarding method to "prevent repeated flooding" in systems using "path learning" and "link aggregation." Ex. 1005, Abstract. Ishimori discloses a system with node groups 60, 70 in between two terminals A10 and B20. *See* Ex. 1005, Fig. 9. The node groups 60, 70 have communication cards #0, #1, #2 and #3, #4, #5, respectively. *Id.* ¶ 15. Ishimori teaches a method where the source MAC address of a received packet is learned by storing the source MAC address "in a buffer [MAC table] had by each node." *Id.* ¶ 2. Thus, the destination MAC address for a packet is searched for "in the MAC table of the local device." *Id.* ¶ 4. If the result is found in the buffer (MAC table),

then the packet is transmitted according to the forwarding path that was learned. *Id.* Thus, Ishimori teaches that "as long as a learning result relating to the destination MAC address (DA) of the receive packet exists in the buffer, there is no need for flooding and traffic in the connected LAN can be kept to a minimum, since the forwarding path to apply to the corresponding packet can be decided unambiguously by using this information." *Id.* ¶ 6.



[FIG. 9]
Block diagram for describing one example of conventional link aggregation function

79. However, if the MAC address is not found, Ishimori acknowledges that in previous approaches, the packet would be flooded to all nodes. *Id.* ¶ 2. Ishimori recognized, however, the inefficiency in this approach because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶ 19. Ishimori then teaches a solution to this flooding

problem. Specifically, Ishimori teaches, by leveraging "link aggregation," which bundles a plurality of ports to function as one virtual port, that "one representative port is selected from among the large number of ports." *Id.* ¶ 13. And the packet is therefore transmitted "using this representative port." *Id.* Thus, "a packet received from the terminal 10 and addressed to terminal 20 is received at card #0" of node 60. *Id.* ¶ 15. "One representative port is selected . . . and this packet is further transmitted." *Id.* Ishimori also teaches that for "each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16. Ishimori also teaches that a "learn packet" is generated at a "predetermined timing." *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33.

80. Ishimori also teaches an aging process. When the buffers of the cards of each node learn a MAC address, the "hit bit" corresponding to the learned MAC address is set to "1." *Id.* ¶ 9. Then, after the first aging cycle, the hit bit is set to "0," which indicates that the MAC address should be deleted from the buffer on the next cycle. If the MAC address is learned again at the next aging process, the hit bit is again set to "1." *Id.*

3. Edsall

81. Edsall is generally directed to techniques for updating and synchronizing "forwarding tables contained on line cards that are interconnected by a switch fabric of a distributed network switch." Ex. 1006 at Abstract. Edsall teaches that the "forwarding table" has an L2 portion that is "used to execute forwarding decision operations for frames forwarded among ports of the line cards." *Id.* at 5:66–6:4. Similar to Smith and Ishimori, in Edsall, "[i]f the frame is received at the ingress card for the first time, this ingress forwarding engine also 'learns' a source MAC address of the frame." *Id.* at 6:26–31. This involves "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 6:31–34. "The ingress forwarding engine then performs a flood-to-fabric (FF) operation on the frame by asserting all bits in the POE mask field of the fabric frame. The asserted POE bits instruct the switch fabric to switch ("flood") copies of the fabric frame through its port interfaces to all (egress) line cards of the network switch." *Id.* at 6:34–39; *see also id.* at 18:44–47.

82. Edsall further teaches a "novel MN frame is provided to complement the FF operation." *Id.* at 6:46–50. The MN (MAC notification) frame "involves use of a primary input (PI) indicator," which "denotes a primary input MAC address that is directly attached to a port of the line card associated with the forwarding table containing this entry." *Id.* at 6:50–56. The "PI indicator is asserted for a forwarding

table entry having a MAC address that is learned from a frame sourced through one of the ports of the line card, as opposed to being learned through the switch fabric." *Id.* at 6:56–60; *see also id.* at 18:56–19:5. The frame additionally includes a POE (port-of-exit) field that "includes a plurality of bits, one for each port interface of the switch fabric." *Id.* at 6:24–25. The POE bit instructs the switch which port interfaces on which line cards should receive the MN frame. *Id.* at 9:47–50. Edsall also teaches an aging process "directed to aging of entries in the forwarding tables of the distributed switch." *Id.* at 17:41–42. "[A] MAC address entry that has not been refreshed as a source within a specified period of time is removed from the L2 portion of the forwarding table in connection with a conventional aging policy executed by the microprocessor on the line card." *Id.* at 17:43–47; *see also id.* at 17:41–18:34.

4. Zelig

83. Zelig is directed to a data communication network that "includes a plurality of primary virtual bridges, interconnected by primary virtual connections." Ex. 1007, Abstract. Zelig discloses that "MAC bridges that implement the 802.1D standard allow MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* ¶ 3. A MAC bridge "includes two or more MAC devices that interconnect the bridge ports to respective LANs." *Id.* The "MAC bridges maintain a database to map destination MAC

address of the packets they receive to bridge ports." *Id.* ¶ 4. "The bridge builds the database by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* Zelig provides "improved mechanisms for protection from failure in virtual private networks (VPNs)." *Id.* ¶ 18. Zelig thus discloses a data communication network including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" that are "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶ 21–24. "[E]ach of the primary and backup virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table." *Id.* ¶ 27. The VPN 20 includes multiple primary virtual bridges. *Id.* ¶ 42.

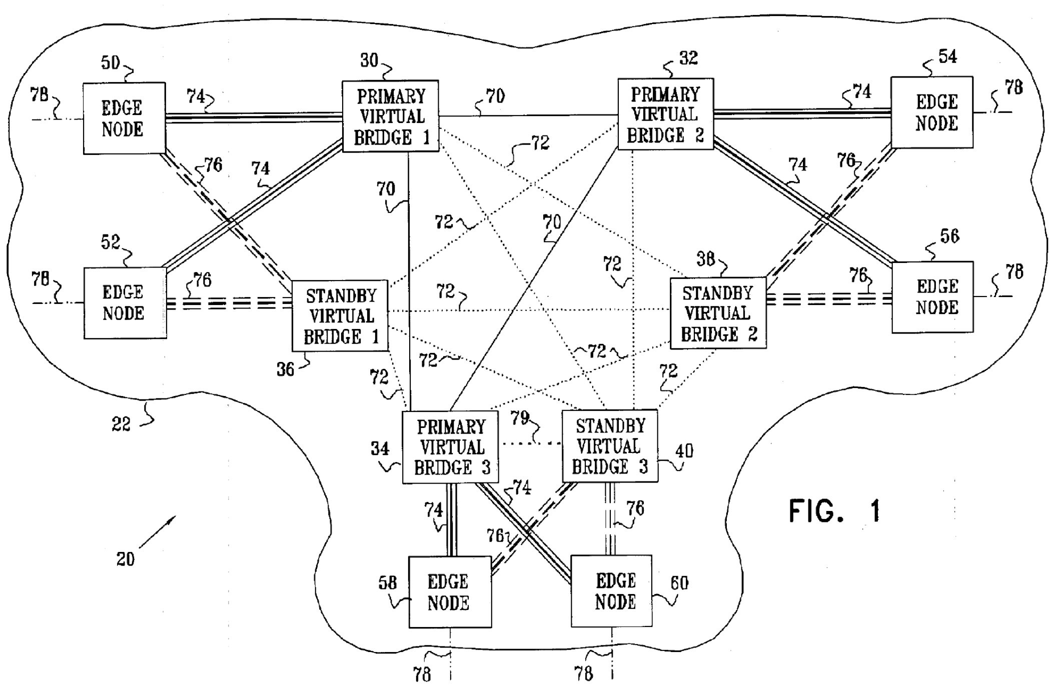


FIG. 1

5. 802.1Q-1998

84. The 802.1Q-1998 standard defines the architecture for Virtual Bridged LANs, its services, protocols, and algorithms. Ex. 1008, Abstract. It was part of the effort of the IEEE to standardize virtual LAN services in bridged LANs. The standard discloses a twelve-bit VLAN identifier field that is used to "uniquely identify the VLAN to which the frame belongs." *Id.* § 9.3.2.3. This standard would have been known to a POSA at the time of the '400 patent.

B. Motivations to Combine

1. Smith and Ishimori

85. A POSA would have been motivated to combine the teachings of Smith and Ishimori, because both references disclose link bundling and path learning techniques in communication networks and sending packets through one port of the link bundle. As explained above, Smith discloses that "[e]ach packet sent between the virtual network device and the first network device is sent via only a one of the communication links." Ex. 1004 ¶ 9. Smith further teaches that "[f]or interfaces (e.g., ports or uplinks) included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet." *Id.* ¶ 66.

86. Similarly, Ishimori also discloses flooding via one representative port. Ex. 1005 ¶ 13. Ishimori recognized the shortcomings in the approach taken by others

of flooding all ports because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶ 19. Ishimori explains that using "one representative port is selected from among the large number of ports had by this trunk according to a predetermined computation, and actual packet transmission is performed using this representative port." *Id.* ¶ 13. "This configuration prevents the application of a different path in the same trunk to the same packet." *Id.* ¶ 14.

87. A POSA looking to solve the shortcomings of flooding all the ports would have therefore looked to Smith and Ishimori. Smith teaches that selecting one egress interface on a virtual network bundle. Ex. 1004 ¶ 66. Ishimori teaches that using "one representative port" prevents constant flooding. A POSA looking to implement this solution would have therefore been motivated to combine their teachings because flooding over only one link in a bundle would prevent sending similar packets unnecessarily over multiple links. A POSA would have understood that flooding would also not occur over the ingress port.

88. Additionally, Ishimori teaches a "learn packet" that is generated at a "predetermined timing." Ex. 1005 ¶ 25. If one of skill in the art were looking to limit flooding over unnecessary links, one would have been motivated to implement Ishimori's learn packet generated at a predetermined timing into Smith's MAC notification method. Furthermore, one of skill in the art would have had a reasonable

expectation of success in implementing this combination because it would have required a simple substitution of Ishimori's structures for Smith's relevant structures. Both Smith and Ishimori leverage link bundling/aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within the knowledge of a POSA implementing such a method to combine the teachings of Smith and Ishimori and have a reasonable expectation of success.

2. Smith, Ishimori, and Edsall

89. A POSA would have been motivated to combine Smith, Ishimori, and Edsall. All three references disclose methods of using MAC forwarding tables in a distributed network switch. Ex. 1004 ¶ 54; Ex. 1005 ¶ 9; Ex. 1006 at Abstract. It would have been obvious to a POSA to modify the MAC forwarding tables of Smith to implement Edsall's forwarding engine. Specifically, a POSA could implement the MAC tables of Smith to implement how Edsall's forwarding engine learns the source MAC address by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." Ex. 1006 at 18:39–44. A POSA could further implement Edsall's "flood-to-fabric (FF) operations" by implementing the MAC tables to include a POE bits that determines which port interfaces on which line cards should receive the MN frames. *Id.* at 18:47–50. Finally, it would have been obvious to a POSA to implement the MAC tables to include the PI indicator as taught in Edsall in order to keep track of whether

a MAC address was learned from a local line card or a different line card. It would have also been obvious to a POSA to further implement the MAC forwarding tables of Smith-Ishimori to use the POE field in order to determine which ports should receive the MN frame.

90. Furthermore, creating entries in a MAC forwarding table and adding a PI indicator field and POE field to the MAC tables would have been an implementation that a POSA would have known how to make with a reasonable expectation of success.

91. Ishimori additionally teaches an aging process. Ishimori's aging process uses a "hit bit." Ex. 1005 ¶ 9. When a MAC address is first learned, the hit bit is set to 1. *Id.* Then, "at the first aging process, this hit bit is cleared," meaning the "hit bit" it is set to 0. *Id.* The learning result is not yet deleted. *Id.* If on the next aging cycle, the MAC address is learned again, the hit bit will be set again to 1. *Id.* Otherwise, the entry will be deleted. *Id.* Therefore, the "hit bit" keeps track of how long a MAC address has been in the MAC address table without having been refreshed. *Id.*

92. A POSA would have understood that MAC addresses may become stale due to changes in the network and would have been motivated to look at solutions for making sure the MAC table is up to date. Although Smith does not disclose aging the entries in the MAC table, it would have been obvious to a POSA to look

for a type of aging parameter could have been used and Ishimori describes precisely that. Moreover, because the size of a MAC table is limited, a POSA would have known that older entries have to be deleted when excessively aged to make space for newer entries. A POSA looking to implement this solution would have therefore been motivated to combine the teachings of Smith and Ishimori and it would have been an easy application of Ishimori's methods to Smith's system. For example, the aging process could be added to the MAC address table described in Smith using known techniques and a simple addition of that information to a table. *Id.*

93. Furthermore, one of skill in the art would have had a reasonable expectation of success in implementing this combination because it would have required a simple implementation of Ishimori's aging process to Smith-Edsall. Smith, Ishimori, and Edsall already leverage link aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within the knowledge of a POSA to simply implement the additional aging process taught by Ishimori to Smith-Edsall, and a POSA would have been able to do so with a reasonable expectation of success. *Id.*

3. Smith, Ishimori, and Zelig

94. A POSA would have been motivated to combine Smith-Ishimori with Zelig. All three references disclose methods of using MAC bridges in a data communication network. Ex. 1004 ¶ 54; Ex. 1005 ¶ 9; Ex. 1007 at Abstract. It

would have been obvious to a POSA to implement the MAC bridges in Smith-Ishimori to maintain a separate MAC table for each VPN MAC bridge as taught in Zelig.

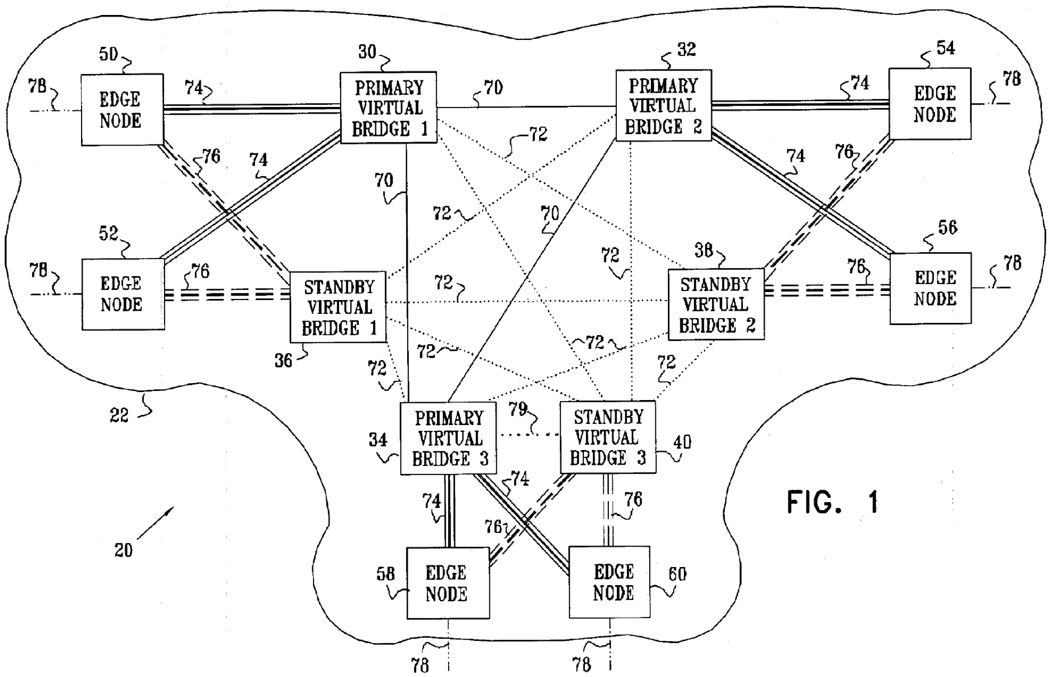


FIG. 1

95. It would have also been obvious to a POSA to configure each virtual MAC bridge to serve a respective VPN, as shown above in Figure 1. Furthermore, implementing the MAC bridges as taught in Zelig to the Smith-Ishimori data communication network would have been an implementation that a POSA would have known how to make with a reasonable expectation of success.

4. Smith, Ishimori, Zelig, and 802.1Q-1998

96. A POSA would have been motivated to combine Smith-Ishimori-Zelig with 802.1Q-1998. All four references pertain to MAC bridges in a data

communication network. Ex. 1004 ¶ 54; Ex. 1005 ¶ 9; Ex. 1007 at Abstract; Ex. 1008 at Abstract. Moreover, Zelig expressly cites to the 802.1Q standard. Ex. 1007 ¶ 12. A POSA would have therefore been motivated to look to the teachings of 802.1Q-1998 in combination with Smith-Ishimori-Zelig to implement the messages from Smith-Ishimori-Zelig with a VLAN identifier in order to identify the VPLS instance. Furthermore, modifying the Smith-Ishimori-Zelig message to add the VLAN identifier would have been a modification that a POSA would have known how to make with a reasonable expectation of success.

C. Specific Grounds of Invalidity

1. Ground 1: Claims 1, 3, 6, 11, 13, and 16 would have been obvious over Smith

97. I have analyzed claims 1, 3, 6, 11, 13, and 16 and in my opinion they would have been obvious over Smith. I provide a detailed analysis of each claim limitation below.

a. Claim 1[pre]: A method for communication, comprising:

98. In my opinion, Smith discloses claim 1[pre]. Smith teaches a system that includes "several communication links." Ex. 1004 ¶ 9. As a stated in the background, computer communication networks are comprised of communication links (and nodes), and these communication links result in a method for communication. Indeed, Smith's disclosure "relates to networking." *Id.* ¶ 2.

b. Claim 1[a]: configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;

99. In my opinion, Smith discloses claim 1[a]. In figure 3 of Smith, there is a **virtual network device 202**, which is a network node. *Id.* ¶ 36. The virtual network device includes **several line cards**, such as 304(1) and 304(4), which may correspond to the first and second line cards. *Id.* ¶ 46. "In virtual network device sub-unit 122(1), line card 304(1) includes forwarding engine 314(1) and interfaces 320(5), 320(7), and 320(9)." *Id.* ¶ 47. "Line card 304(4) includes forwarding engine 314(4) and interfaces 320(12), 320(14), and 320(16)." *Id.* ¶ 48. The interfaces on line cards 304(1) and 304(4) are a plurality of ports. An "interface" and "port" are synonymous and Smith treats them interchangeably, and thus, discloses the first and second ports. *Id.* ¶ 63 (stating "port *or* uplink interface"). Therefore, it is my opinion that Smith's virtual network device, which is a network node, is configured to have a first and second line card with respective first and second ports.

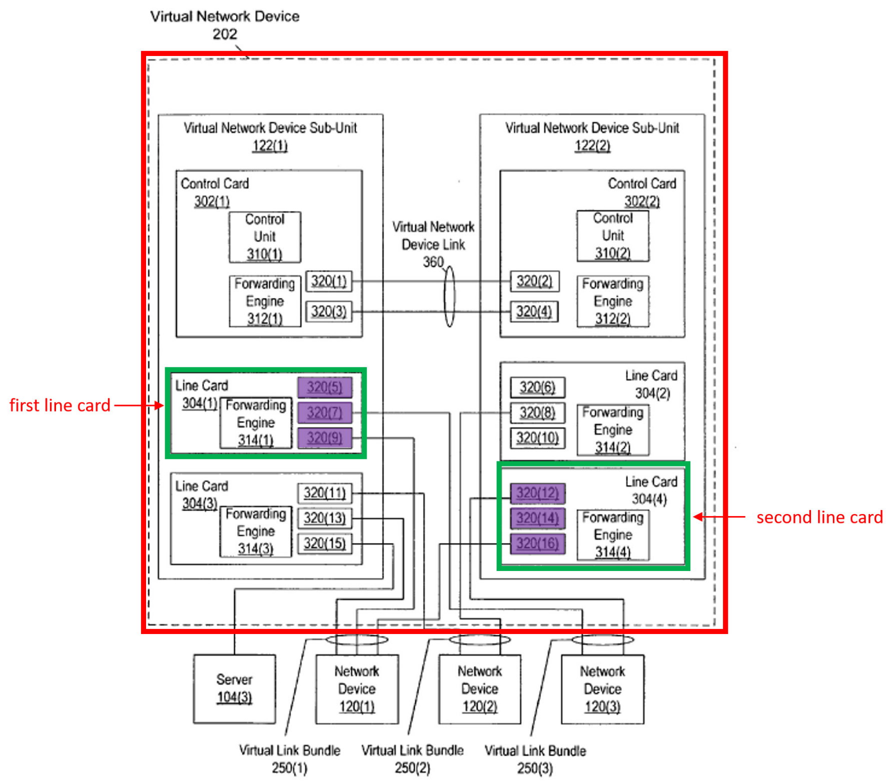


FIG. 3

100. Smith further teaches virtual link bundles, which may be link aggregation (LAG) groups and may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶ 6, 9, 30. Indeed, Smith discloses that:

a. "One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using a **link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**." *Id.* ¶ 6 (emphasis added).

b. "In some embodiments, virtual link bundles **250(1)** and **250(2)** are each operated as an EtherChannel (TM) or as an **aggregated link (as described in IEEE 802.3)**." *Id.* ¶ 35. (emphasis added).

c. "[…] the non-satellite network devices provide **L2 (Layer 2)** and L3 (Layer 3) **forwarding** and routing […]" and "[…] the satellite network devices simply forward all packets to non-satellite network devices for **L2 forwarding** and L3 routing." *Id.* ¶ 30 (emphasis added).

d. "For a given virtual link bundle, that virtual link bundle can be managed (e.g., with respect to control protocols such as **L2 protocols**) in a central location." *Id.* ¶ 57 (emphasis added).

e. "In some embodiments, **MAC** notification frames are used to keep the content of the **L2 tables** in virtual network device sub-unit **122(1)** synchronized with the content of the **L2 tables** in virtual network device sub-unit **122(2)** and vice versa." *Id.* ¶ 62 (emphasis added).

f. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from

a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device sub-unit **122(1)** can learn that the sending device's **MAC (Media Access Control)** address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13).** Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well as each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a **packet addressed to that MAC address can be forwarded** via any of uplink interfaces **320(9), 320(13),** and **320(16).**" *Id.* ¶ 54 (emphasis added).

g. "The communication links are configured to be **managed as a single link**." *Id.* ¶ 9 (emphasis added).

h. "Network devices **120(1)-120(3)** each operate their multiple uplinks to virtual network device **202** as a **single logical uplink**. Additionally, in some embodiments, each network device **120(1)-120(3)** can operate as if that network device is coupled to a **single distribution-layer device**, virtual network device **202,** instead of operating as if that network device were coupled to two independent distribution-layer network devices." *Id.* ¶ 44 (emphasis added).

101. The virtual network device 202 "route[s] and forward[s] packets to and from network devices 120(1)–120(3)" by associating the MAC address of a received data packet with the logical identifier of the uplink interface. *Id.* ¶ 54. Therefore, it is my opinion that Smith's virtual network device 202 operates as a distributed MAC bridge in Layer 2 of the data network.

102. In conclusion, it is my opinion that Smith teaches configuring a network node (the virtual network device) having at least a first and second line card (the line cards 304(1) and 304(4)) with a plurality of ports (the interfaces on the line cards) that operates as a distributed MAC bridge in Layer 2 of the data network.

c. Claim 1[b]: configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined Member Line Cards;

103. In my opinion, Smith discloses claim 1[b]. Smith teaches virtual link bundles, which may be link aggregation (LAG) groups and may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶ 6, 9, 30. Indeed, Smith discloses that:

a. "One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using a **link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**." *Id.* ¶ 6 (emphasis added).

b. "In some embodiments, virtual link bundles **250(1)** and **250(2)** are each operated as an EtherChannel (TM) or as an **aggregated link (as described in IEEE 802.3)**." *Id.* ¶ 36 (emphasis added).

c. "[T]he non-satellite network devices provide **L2 (Layer 2)** and L3 (Layer 3) **forwarding** and routing" and "the satellite network devices simply forward all packets to non-satellite network devices for **L2 forwarding** and L3 routing." *Id.* ¶ 30 (emphasis added).

d. "For a given virtual link bundle, that virtual link bundle can be managed (e.g., with respect to control protocols such as **L2 protocols**) in a central location." *Id.* ¶ 57 (emphasis added).

e. "In some embodiments, **MAC** notification frames are used to keep the content of the **L2 tables** in virtual network device sub-unit **122(1)** synchronized with the content of the **L2 tables** in virtual network device sub-unit **122(2)** and vice versa." *Id.* ¶ 62 (emphasis added).

f. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device sub-unit **122(1)** can learn that the sending device's **MAC (Media Access Control)** address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13).** Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well as each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case,

uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a **packet addressed to that MAC address can be forwarded** via any of uplink interfaces **320(9), 320(13),** and **320(16).** *Id.* ¶ 54 (emphasis added).

g. "The communication links are configured to be **managed as a single link**." *Id.* ¶ 9 (emphasis added).

h. "Network devices **120(1)-120(3)** each operate their multiple uplinks to virtual network device **202** as a **single logical uplink**. Additionally, in some embodiments, each network device **120(1)-120(3)** can operate as if that network device is coupled to a **single distribution-layer device**, virtual network device **202,** instead of operating as if that network device were coupled to two independent distribution-layer network devices." *Id.* ¶ 44 (emphasis added).

104. Smith discloses that **network device 120(2)** or one of the endpoints "is coupled to **virtual network device 202**" or a second endpoint "by **virtual link bundle 250(2)**" as shown in the annotated figure below. *Id.* ¶ 44. The virtual link bundle 250(2) consists of **two uplinks**, which is a plurality of LAG ports. *Id.* ¶ 51.

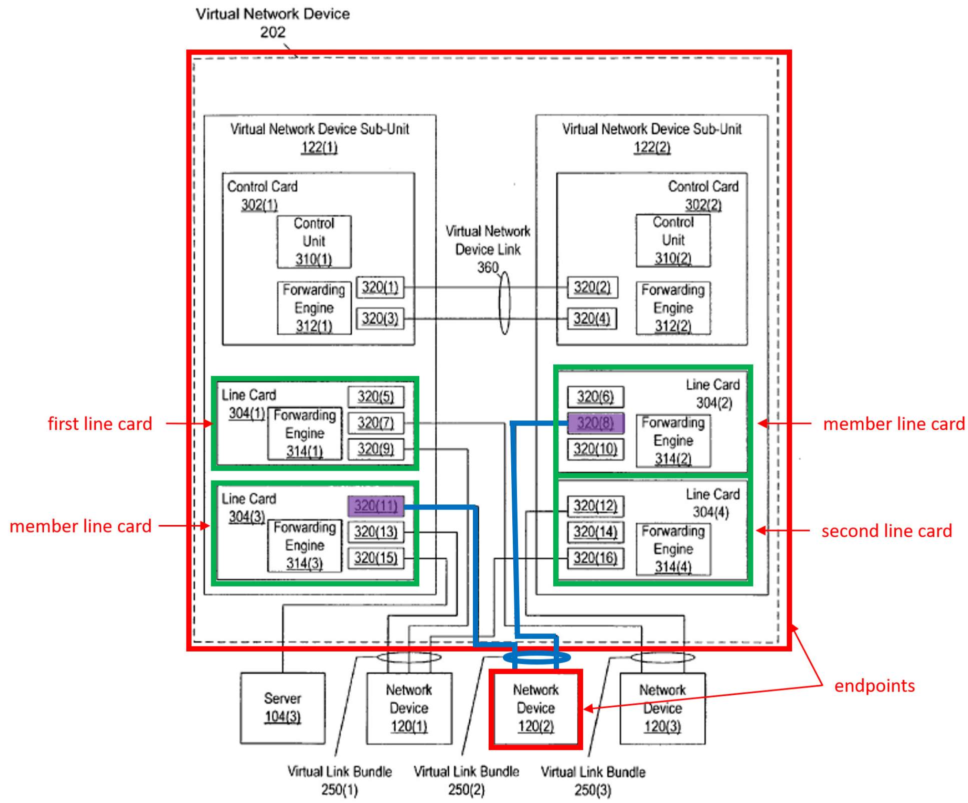


FIG. 3

105. Smith further specifies that the virtual network devices "provide L2 (Layer 2) . . . forwarding and routing." *Id.* ¶ 30. Continuing with my above example, line cards 304(1) and 304(4) may correspond to the first and second line card respectively. The other two line cards 304(2) and 304(3) may make up the conjoined member line cards, because all four line cards are part of the virtual network device sub-units 122(1) and 122(2) that "can coordinate their behavior such that they appear to be a single virtual device." *Id.* ¶ 50. In my opinion, Smith therefore discloses configuring a link aggregation (LAG) group of parallel physical links (the virtual link bundle 250(2)) between two endpoints (virtual network device 202 and network

device 120(2)) in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports (interfaces 320(8) and 320(11)) and a plurality of conjoined Member Line Cards (line cards 304(2) and 304(3)).

d. Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

106. In my opinion, Smith discloses claim 1[c]. The member line cards in Smith include **interfaces**, which are ports. Ex. 1004 ¶ 47. And the forwarding tables in Smith correspond to forwarding databases. Indeed, Smith discloses:

a. "Line card **304(3)** includes forwarding engine **314(3), interfaces 320(11)** and **320(13),** and **port 320(15).** *Id.* ¶ 47 (emphasis added).

b. "identifier of the identified **port or uplink interface** . . . " *Id.* ¶ 65 (emphasis added).

c. "If the packet is received from a local uplink **interface or port** (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405,** the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can provide the destination address to a **forwarding table** in order to determine which **logical identifier, if any, is associated with that destination address**. If there is no hit

in the **forwarding table**, as determined at **407,** the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409.** For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

d. "If the packet's destination address hits in the **forwarding table**, as determined at **407,** the virtual network device sub-unit uses the logical identifier returned by the **forwarding table** to select the interface(s) to which the packet should be sent. If the **forwarding table** does not identify an interface bundle, as determined at **411**, the packet is sent via the identified port(s) and/or uplink

interface(s), as indicated at **413.** If the **forwarding table** does identify an interface bundle, the virtual network device sub-unit sends the packet via one local interface included within the identified interface bundle, as shown at **415** (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

e. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the packet's destination device. In this example, this is performed by providing the destination address to a **forwarding table**, as shown a **417.** If there is not a hit in the **forwarding table** (i.e., if no association has already been learned for the destination address), the virtual network device sub-unit floods the packet on the incoming VLAN." *Id.* ¶ 69 (emphasis added).

f. "If there is a hit in the forwarding table, and if the forwarding table does not identify an interface bundle at **421,** the packet is sent via the identified port and/or uplink interfaces, as indicated at **423.** If instead the forwarding table does identify an interface bundle, the packet is not sent via that interface bundle." *Id.* ¶ 70 (emphasis added).

g. "When virtual network device sub-unit **122(1)** looks up the destination address of the packet in a **lookup table**, the lookup table returns the logical identifier that identifies local uplink interfaces **320(9)** and **320(13).** The packet is then forwarded to uplink interface **320(13)** (e.g., selected based on load-sharing considerations)." *Id.* ¶ 61 (emphasis added).

h. "For example, control unit **310(2)** can use this information to set up or modify **lookup tables on line cards 304(2)** and/or **304(4).**" *Id.* ¶ 57 (emphasis added).

i. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device sub-unit **122(1)** can learn that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13).** Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-

unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).”** *Id.* ¶ 54 (emphasis added).

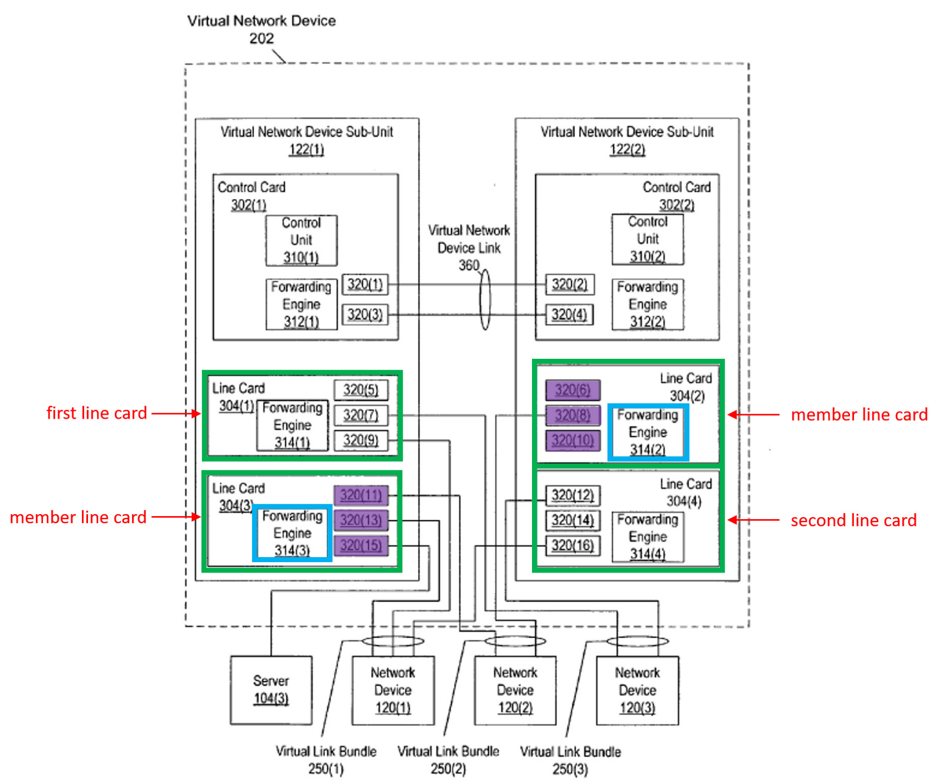


FIG. 3

107. When a packet is received on a particular uplink interface, the virtual network device, which is a network node as described with respect to claim 1[a], learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] uplink interface" or the ports of said plurality of ports. *Id.* ¶ 54. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device 202 to route and forward packets to and from network devices 120(1)-120(3)." *Id.* Smith teaches that this information may be used to "set up or modify lookup tables," which maps to a forwarding database (FDB). *Id.* ¶¶ 57, 61. Therefore, it is my opinion that Smith teaches that the lookup table or forwarding table or FDB may store records associating MAC addresses with the logical identifier of the uplink interface of said network node.

e. Claim 1[d]: receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;

108. In my opinion, Smith discloses claim 1[d]. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a **packet from a sending device (e.g., a client coupled to network device**

120(1)) is received via uplink interface 320(13), virtual network device sub-unit 122(1) can learn that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13).** Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well **as** each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, **packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier**. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).**" *Id.* ¶ 54 (emphasis added)

b. "If the packet is **received from a local uplink** interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405,** the virtual network device sub-unit attempts to **forward the**

packet to its destination address. For example, the virtual network device sub-unit can provide the **destination address** to a forwarding table in order to determine which logical identifier, if any, is associated with that **destination address**. If there is no hit in the forwarding table, as determined at **407,** the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409.** For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

109. Smith discloses that the uplink interface receives a data packet with the "sending device's MAC address." *Id.* ¶ 54. The uplink interface is an ingress port

and the sending device's MAC address is the MAC source address. For example, if the network device 120(1) is the MAC source address, the ingress port would be interface 320(9) from a data packet received from network device 120(1).

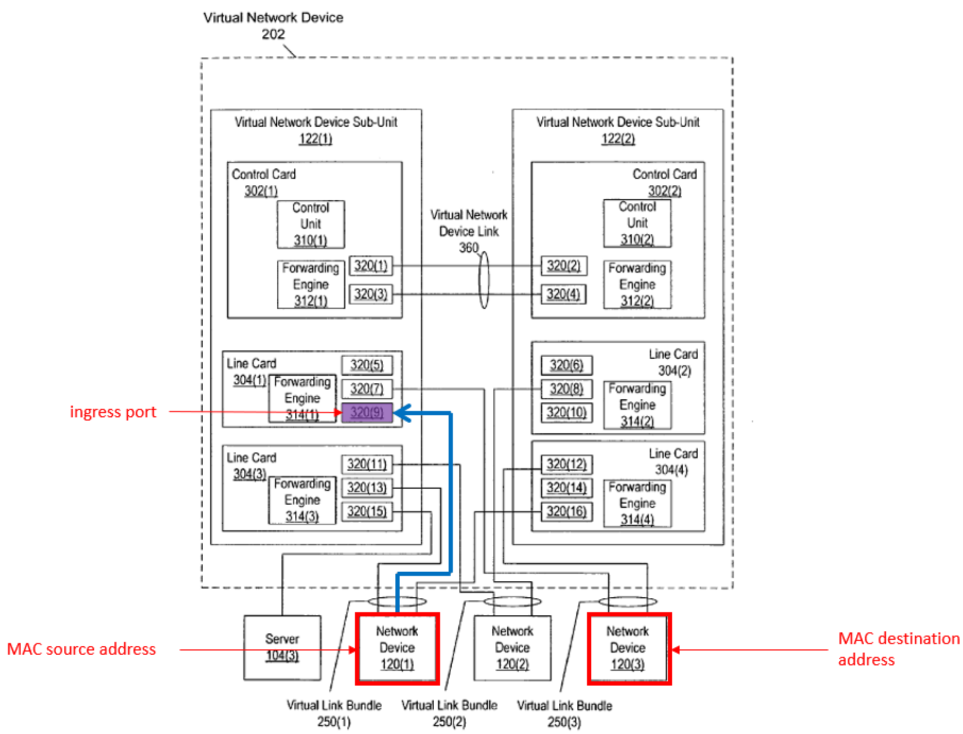


FIG. 3

110. It is therefore my opinion that Smith teaches receiving a data packet on an ingress port (interface 320(9)) of said network node (virtual network device 202) from a MAC source address (network device 120(1)).

111. Smith further discloses that this data packet has a "destination logical identifier." *Id.* ¶¶ 60, 62. The MAC destination address could be network device 120(3), for example. Smith specifies that these devices provide Layer 2 forwarding. *Id.* ¶ 30. The data packet's MAC destination address (network device 120(3)) is on

the Layer 2 of the data network. In my opinion, Smith teaches receiving a packet on an ingress port (for example, interface 320(9)) of said network node from a MAC source address (for example, network device 120(1)), said data packet specifying a MAC destination address (for example, network device 120(3)).

f. Claim 1[e]: conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;

112. In my opinion, Smith discloses claim 1[e]. Smith discloses:

a. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a **packet from a sending device (e.g., a client coupled to network device 120(1)) is received via uplink interface 320(13), virtual network device sub-unit 122(1) can learn that the sending device's MAC (Media Access Control) address** is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13).** Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well **as** each forwarding engine in virtual network

device sub-unit **122(2)** of this association. Based on the association, **packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier**. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).”** *Id.* ¶ 54 (emphasis added).

b. “If the packet is **received from a local uplink** interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405,** the virtual network device sub-unit attempts to **forward the packet to its destination address**. For example, the virtual network device sub-unit can provide the **destination address** to a forwarding table in order to determine which logical identifier, if any, is associated with that **destination address**. If there is no hit in the forwarding table, as determined at **407,** the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation,

> the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409.** For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one **egress interface** per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

113. Smith teaches that the data packet is forwarded based on "the association between a packet and a particular logical identifier." *Id.* ¶ 54. Smith additionally teaches that its system "favor[s] local interfaces." *Id.* ¶ 56. Again, referring to Figure 3, in order to transmit the data packet to the MAC destination address (network device 120(3)), the data packet would be conveyed from interface 320(9) to interface 320(7) (said first port), because the system favors local interfaces. Interface 320(7), which is on line card 304(1) (said first line card), would then transmit the data packet to the MAC destination address (port of network device 120(3)).

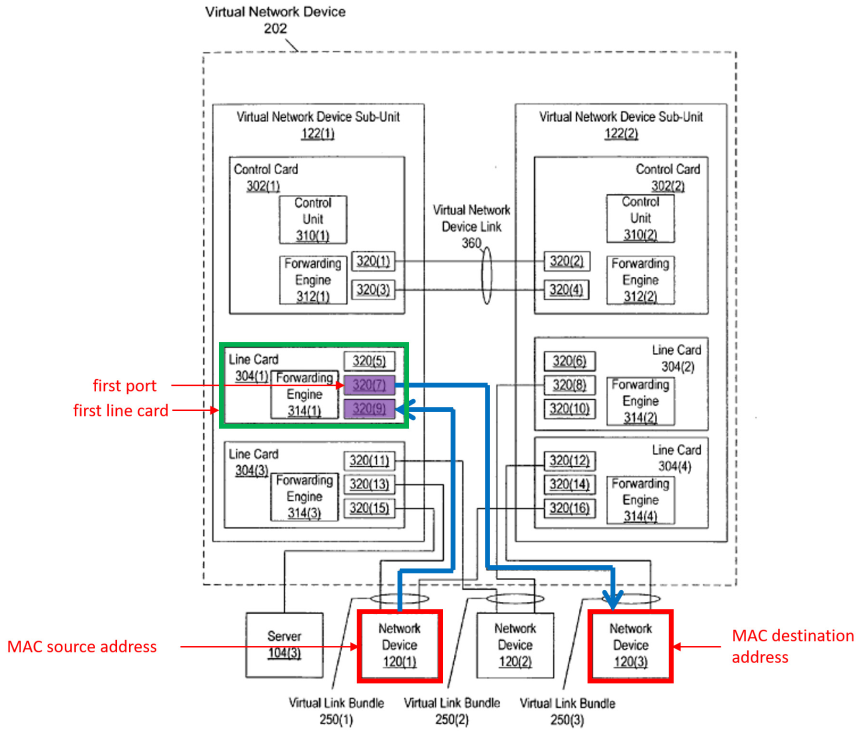


FIG. 3

g. Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;

114. In my opinion, Smith discloses claim 1[f]. Smith teaches:

a. "If the packet is received from a local uplink interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at 405, the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can provide the destination address to a forwarding table in order to determine which logical identifier, if any, is associated

with that destination address. **If there is no hit in the forwarding table**, as determined at 407, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit **floods the packet to all egress ports** and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at 409. For interfaces (e.g., ports or uplinks) **included in interface bundles**, the virtual network device sub-unit selects **one egress interface per interface bundle** via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

b. "If the packet's destination address hits in the forwarding table, as determined at 407, the virtual network device sub-unit uses the logical identifier returned by the forwarding table to select the interface(s) to which the packet should be sent. If the forwarding table does not identify an interface bundle, as determined at 411,

the packet is sent via the identified port(s) and/or uplink interface(s), as indicated at 413. If the forwarding table does identify an interface bundle, the virtual network device sub-unit sends the packet via **one local interface** included within the identified interface bundle, as shown at 415 (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

c. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the packet's destination device. In this example, this is performed by providing the destination address to a forwarding table, as shown a **417. If there is not a hit in the forwarding table** (i.e., if no association has already been learned for the destination address), the virtual network device sub-unit **floods the packet on the** incoming VLAN." *Id.* ¶ 69 (emphasis added).

115. Smith teaches that the virtual network device sub-unit 122(1) looks up the destination address in a "lookup table" (FDB). *Id.* ¶ 61. Smith discloses sharing entries of the FDB with all the member line cards. *Id.* ¶ 63. "If a forwarding engine within virtual network device sub-unit 122(2) already knows that the destination address is behind a particular uplink interface or port . . . , that forwarding engine

generates a MAC notification identifying this association, which is distributed to any other forwarding engines within virtual network device sub-unit 122(2)." *Id.* It is my understanding therefore that one way to read "said FDB" in the claim is that it could be referring to the FDBs associated with any of the member line cards. Alternatively, "said FDB" could refer to the FDB on the line card associated with the ingress port, which would be line card 304(1) or the first line card in my example above. Although there is no line card or FDB associated with the ingress port or first line card recited in the claim, it would have been obvious to a POSA that the first line card could have an FDB.

116. If there is no hit in the forwarding table, "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN." *Id.* ¶¶ 63, 66. Smith discloses that "flooding" is "via one and only one LAG port." In particular, Smith teaches that when the network device sends a packet "via the virtual link bundle," it "selects one of the communication links on which to send the packet." *Id.* ¶ 9. Thus, it my opinion that Smith discloses, or at least renders obvious, sending the data packet via one and only one LAG port.

h. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

117. In my opinion, Smith discloses claim 1[g]. Smith teaches:

a. "If the packet is received from a local uplink interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405,** the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can **provide the destination address to a forwarding table** in order to determine which **logical identifier, if any, is associated with that destination address**. If there is no hit in the forwarding table, as determined at **407,** the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409.** For interfaces (e.g., ports or uplinks) included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all

interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

b. "If the packet's destination address **hits in the forwarding table**, as determined at **407,** the virtual network device sub-unit uses the **logical identifier returned by the forwarding table** to select the interface(s) to which the packet should be sent. If the forwarding table does not identify an interface bundle, as determined at **411**, the packet is sent via the identified port(s) and/or uplink interface(s), as indicated at **413.** If the forwarding table does identify an interface bundle, the virtual network device sub-unit sends the packet via one local interface included within the identified interface bundle, as shown at **415** (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

c. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the packet's destination device. In this example, this is performed by **providing the destination address to a forwarding table**, as shown a **417.** If there is not a hit in the forwarding table (i.e., if no association has already been learned for the destination address),

the virtual network device sub-unit floods the packet on the incoming VLAN." *Id.* ¶ 69 (emphasis added).

d. "If there is a **hit in the forwarding table**, and if the forwarding table does not identify an interface bundle at **421,** the packet is sent via the identified port and/or uplink interfaces, as indicated at **423.** If instead the forwarding table does identify an interface bundle, the packet is not sent via that interface bundle." *Id.* ¶ 70 (emphasis added).

e. "When virtual network device sub-unit **122(1) looks up the destination address of the packet in a lookup table**, the lookup table returns the logical identifier that identifies local uplink interfaces **320(9)** and **320(13).** The packet is then forwarded to uplink interface **320(13)** (e.g., selected based on load-sharing considerations)." *Id.* ¶ 61 (emphasis added).

118. Smith teaches that "[b]ased on the source address of the packet and which port or uplink interface received the packet, the virtual network device sub-unit learns the source identifier of the sending device, as indicated at 403." *Id.* ¶ 65. These identifiers are stored in a lookup table on a virtual network device. *Id.* ¶ 61. It would have therefore been obvious to check the MAC source address in

the records of the FDB of said first line card (line card 304(1)) to determine if it exists in the FDB.

i. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.

119. In my opinion, Smith discloses claim 1[h]. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13).** Virtual network device sub-unit **122(1)** **can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the

association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).”** *Id.* ¶ 54 (emphasis added).

b. “In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device sub-unit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification** can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via** to virtual network device link **360.** Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will

send a copy of that packet via virtual network device link **360,** ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

c. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device sub-unit **122(1)** sends a copy of the packet to virtual network device sub-unit **122(2)** via virtual switch link **360.** If a forwarding engine within virtual network device sub-unit **122(2)** already knows that the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120),** that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device sub-unit **122(2).** Since the packet was originally received via virtual

network device link **360,** virtual network device sub-unit **122(2)** also sends a copy of the MAC notification back via virtual network device link **360.** This MAC notification can then be distributed among the forwarding engines included in virtual network device sub- unit **122(1).** After being updated based on the MAC notifi cation, the forwarding engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently- received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

120. A POSA would have understood that if the record of an association between the MAC source address with said ingress port did not exist in the lookup table of, for example, line card 304(1) (said first line card), it would create a new entry and add it to the lookup table of line card 304(1). Smith teaches that the network device sub-unit 122(2) sends a MAC notification (a message) to update the forwarding engines when it learns of a new association. *Id.* ¶ 63. "After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand this to mean that the MAC notification,

which is a message, is therefore sent to each member line card (304(2) through 304(4)).

j. Claim 3: The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

121. In my opinion, Smith discloses claim 3. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13).** Virtual network device sub-unit **122(1)** **can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent

from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).**" *Id.* ¶ 54 (emphasis added)

b. "In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device sub-unit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification** can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via** to virtual network device link **360.** Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will send a copy of that packet via virtual network device link **360,**

ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

c. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device sub-unit **122(1)** sends a copy of the packet to virtual network device sub-unit **122(2)** via virtual switch link **360.** If a forwarding engine within virtual network device sub-unit **122(2)** already knows that the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120),** that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device sub-unit **122(2).** Since the packet was originally received via virtual network device link **360,** virtual network device sub-unit **122(2)**

> also sends a copy of the MAC notification back via virtual network device link **360.** This MAC notification can then be distributed among the forwarding engines included in virtual network device sub- unit **122(1).** After being updated based on the MAC notifi cation, the forwarding engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently- received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

122. Smith discloses that the network device sub-unit 122(2) sends a MAC notification (the message) to update the forwarding engines. *Id.* ¶ 63. Smith specifies that "[a]fter being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand that this includes the other line cards in virtual network device 202, such as line card 304(4). For example, line card 304(4) (*e.g.*, the second line card) would receive the MAC notification (the message) and update its lookup tables. *Id.* ¶ 57. Thus, Smith discloses that in response to the message, the record of association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

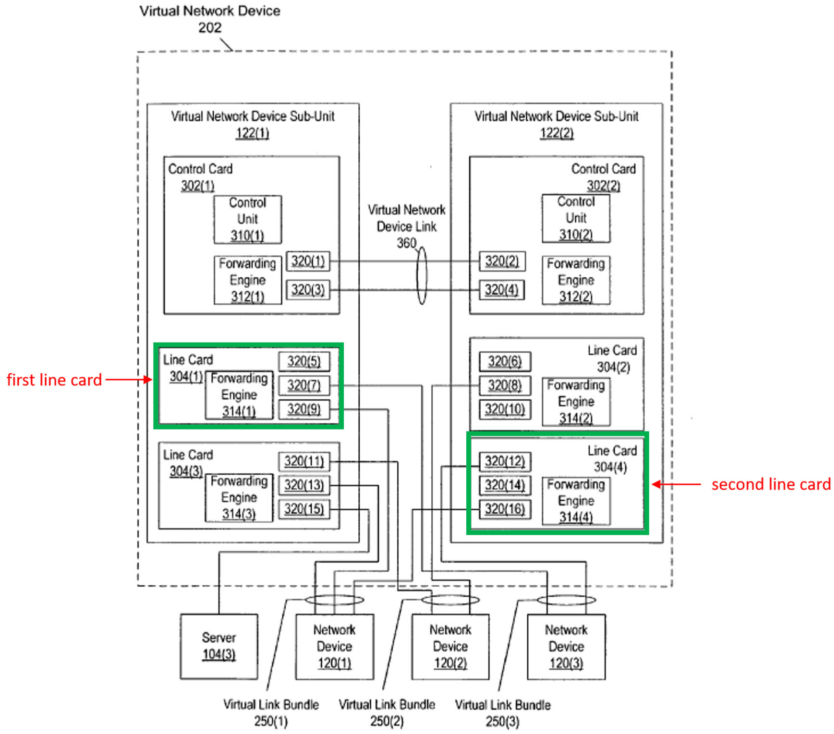


FIG. 3

k. Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via switching core of the network node to at least the second line card.

123. In my opinion, Smith discloses claim 6. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3).** For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13),** virtual network device

sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13).** Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13),** a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16).”** *Id.* ¶ 54 (emphasis added).

b. “In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device sub-unit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle

or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification** can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via** to virtual network device link **360.** Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will send a copy of that packet via virtual network device link **360,** ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

c. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device sub-unit **122(1)** sends a copy of the packet to virtual network device sub-unit **122(2)** via virtual switch link **360.** If a forwarding engine within virtual network device sub-unit **122(2)** already knows that

the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120),** that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device sub-unit **122(2).** Since the packet was originally received via virtual network device link **360,** virtual network device sub-unit **122(2)** also sends a copy of the MAC notification back via virtual network device link **360.** This MAC notification can then be distributed among the forwarding engines included in virtual network device sub- unit **122(1).** After being updated based on the MAC notifi cation, the forwarding engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently- received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

124. The line cards communicate with each other in order to transfer various messages between them. For example, note that "such a notification can be generated by one line card in order to update an L2 table on another line card." *Id.* ¶ 62.

Therefore, the line cards must communicate, allowing for sending messages from one line card to another. This, internal to device 202, switching mechanism allowing the line cards to communicate with each other could be the switching core.

125. Smith teaches that the "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* Thus, it would have been obvious to a POSA to implement the MAC notification frame as a synchronization packet and transmit it to at least the second line card via the switching core.

l. Claim 11[pre] 11: A node for network communication, comprising:

126. In my opinion, Smith discloses claim 11[pre]. Smith teaches that its system contains communications links that are coupled to network devices. *Id.* ¶ 9. A POSA would understand that a network device is a node.

m. Claim 11[a]: a switching core;

127. In my opinion, Smith discloses claim 11[a]. It would be within the knowledge of a POSA that the line cards in Smith would be communicating with each other, as discussed before. *See supra* ¶ 124.

n. Claim 11[b]: a plurality of ports;

128. In my opinion, Smith discloses claim 11[b]. The network devices in Smith contain several ports (a plurality of ports). Ex. 1004 ¶ 6. "For example, an

EtherChannel (TM) port bundle can be formed ***from several ports on a switch***, each of which is coupled to a respective link in a group of links coupling that switch to another switch." *Id.*

o. Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

129. In my opinion, Smith discloses claim 11[c]. Claim 11[c] combines the elements of claims 1[a], 1[b], 1[c], and 11[a]. Therefore, for the same reasons that the Smith discloses claims 1[a], 1[b], 1[c], and 11[a], Smith also discloses claim 11[c]. *See supra* ¶¶ 99–107, 127.

p. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds

said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

130. In my opinion, Smith discloses claim 11[d]. This limitation is a combination of claims 1[d], 1[e], 1[f], 1[g], and 1[h]. Therefore, for the same reasons that Smith discloses claims 1[d], 1[e], 1[f], 1[g], and 1[h], Smith also discloses claim 11[d]. *See supra* ¶¶ 108–120.

q. Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.

131. In my opinion, Smith discloses claim 13. *See supra* ¶¶ 121–122.

r. Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

132. In my opinion, Smith discloses claim 16. *See supra* ¶¶ 123–125.

2. Ground 2: Claims 1–3 and 11–13 would have been obvious over Smith in view of Ishimori

133. I have analyzed claims 1–3 and 11–13 and conclude that they would have been obvious over Smith in combination with Ishimori. I provide a detailed analysis of each claim limitation below.

a. Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

134. In my opinion, the combination of Smith and Ishimori render obvious claim 1[c]. As explained above, when a packet is received on a particular uplink interface, the virtual network device learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] uplink interface." Ex. 1004 ¶ 54. Smith teaches that the information learned about the association between a packet and a particular logical identifier may be used to "set up or modify lookup tables" (*e.g.*, the forwarding database (FDB)). *Id.* ¶¶ 54, 57, 61, 71.

135. To the extent Smith alone does not render obvious claim 1[c], Smith in combination with Ishimori does. Indeed, Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of learning a source MAC address of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets, an IP packet having an unlearned MAC address as a destination address, a transmission path thereof cannot be specified. As such, in this situation, a broadcast transmission to all nodes (that is,

"flooding") is performed. Then, at this time, path **learning** is carried out **by storing** the source MAC address (SA) had by this IP packet and a port that received such in a buffer had by each node." Ex. 1005 ¶ 2 (emphasis added).

b. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is supposed wherein each node—50-0, 50-1, 50-2—has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A", which is a source MAC address (SA) thereof, is learned by being **stored** in a predetermined buffer in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the **MAC table**, this information is overwritten." *Id.* ¶ 3 (emphasis added).

c. "Next, for a destination address (DA; here, MAC address "B") had by this receive IP packet, it is searched whether there is a learning result in the MAC table of the local device (step S3). Then, when the search result is a mishit ("No"), this IP packet is broadcast

(flooded) to all nodes (in this example, node 50-1 and node 50-2) (steps S5, P2). Meanwhile, when, as a result of the search at step S3, a learning result relating to the corresponding destination MAC address exists in the buffer ("Yes"), this IP packet is transmitted according to a forwarding path—that is, a card number and port number—included in this learning result (step S4). In this situation, there is no need for flooding." *Id.* ¶ 4 (emphasis added).

d. "Note that the same learning operations are also performed when this IP packet sent from node 50-0 reaches card #2 of node 50-2, which the IP packet passes through on its way to a corresponding destination terminal B 20. As a result, as illustrated in FIG. 1, source path information at this time—that is, the information on card #0, port #0 of node 50-0—is stored as a learning result in association with the MAC address "A" of the source terminal 10 **in the buffer** (**MAC table**) of this card #2." *Id.* ¶ 5 (emphasis added).

136. Ishimori teaches that each of its communication cards has a "MAC table." *Id.* ¶ 2. Referring to Figure 1, Ishimori explains that when a packet from terminal A is received at port #0 of card #0, the path information for address "A" is learned by storing the information in the MAC table "had by this card" (*e.g.*, card

#0). *Id.* ¶ 3. Ishimori therefore teaches that its line cards maintain a MAC table to store path information of the MAC address and the receiving card and port. *Id.* In other words, the MAC table holds records associating MAC addresses with ports of said plurality of ports of said network node. It would have been obvious to substitute the line cards from Ishimori into Smith and a POSA would have had a reasonable expectation of success in making this simple substitution.

b. Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;

137. In my opinion, to the extent Smith does not explicitly teach the limitation of sending the packet to one and only one LAG port, Ishimori expressly discloses flooding via "one representative port" (*e.g.*, via one and only one LAG port).

a. "Next, a **link aggregation** function is described. This link **aggregation function is a function of using a plurality of ports by deeming such to be one virtual high-speed port**. This function enables a band improvement effect to be substantially obtained. The **virtual port in this situation is referred to as a trunk**. In using this trunk (referred to as "trunking"), **when, as above, flooding is to be performed when an address is not learned**, **one representative port is selected** from among the large

number of ports had by this trunk according to a predetermined computation, and **actual packet transmission is performed using this representative port**." *Id.* ¶ 13 (emphasis added).

b. "Then, at a node that receives an IP packet via the trunk, when carrying out path learning of a source address of this packet, instead of learning the port number at the time of receipt, information on the trunk including such is stored as the learning result. Afterward, in receiving an IP packet having a destination that passes through this trunk, one path is decided from among the paths constituting the trunk according to predetermined computation methods, independent of each other according to hardware, of these communication cards, and this decided path is applied to the corresponding packet. This configuration **prevents the application of a different path in the same trunk to the same packet**." *Id.* ¶ 14 (emphasis added).

138. Ishimori teaches that when "flooding is to be performed when an address is not learned, one representative port is selected from among the large number of ports had by this trunk according to a predetermined computation, and actual packet transmission is performed using this representative port." *Id.* ¶ 13. One of ordinary skill in the art would have been motivated to combine the teachings

of Smith with Ishimori because Ishimori explains that using "one representative port . . . prevents the application of a different path in the same trunk to the same packet." *Id.* ¶¶ 13, 14. Thus, a POSA would have looked to both Smtih and Ishimori, which explain why it would be beneficial to use only one representative port, and combine the teachings of the two references as they are directed to similar methods. Furthermore, a POSA would have had a reasonable success in combining Smith with Ishimori because it would have been a simple application of Ishimori's methods on Smith's virtual network device. Thus, the combination of Smith and Ishimori renders obvious claim 1[f].

c. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

139. In my opinion, to the extent this is not disclosed by Smith, Ishimori teaches that the source MAC address is stored with the reception path information including the card number and port number. *Id.* ¶ 3. Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of **learning a source MAC address** of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets,

an IP packet having an unlearned MAC address as a destination address, a transmission path thereof cannot be specified. As such, in this situation, a broadcast transmission to all nodes (that is, "flooding") is performed. Then, at this time, path **learning** is carried out **by storing** the source MAC address (SA) had by this IP packet and a port that received such in a buffer had by each node." *Id.* ¶ 2 (emphasis added).

b. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is supposed wherein each node—50-0, 50-1, 50-2—has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A", which is a source MAC address (SA) thereof, **is learned** by being **stored** in a predetermined buffer in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the **MAC table**, this information is overwritten." *Id.* ¶ 3 (emphasis added).

140. It would have been obvious to a POSA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.* Thus, the combination of Smith and Ishimori renders obvious this limitation.

d. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.

141. In my opinion, to the extent Smith alone does not render obvious this limitation, in my opinion, it would have been obvious to combine Smith with Ishimori. Ishimori discloses that the source MAC address is stored with the reception path information including the card number and port number. *Id.* ¶ 3. Indeed, Ishimori's disclosures include:

a. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is supposed wherein each node—50-0, 50-1, 50-2—has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A",

which is a source MAC address (SA) thereof, is learned by being **stored in a predetermined buffer** in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the MAC table, this information is overwritten." *Id.* ¶ 3 (emphasis added).

b. "The path having this representative port leads to card #3 of the nodes 70. At this card #3 as well, no learning result relating to the terminal 20 that is the destination is had. As such, at card #3 as well, this packet is flooded (step P13). In this manner, this packet reaches the terminal 20 that is the destination through card #5. Note that during the packet forwarding from the terminal 10 to the terminal 20 illustrated in FIG. 9 and FIG. 10, at each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16.

c. "According to the present invention, each packet forwarding device is configured to **generate a learn packet at a predetermined timing under predetermined conditions**. That is,

even in a situation wherein different paths are selected for a coming direction and a going direction due to trunking or the like and thus only forwarding in one of these directions is performed in a certain device, by generating a learn packet at a predetermined timing, this learn packet can force, for example, the device performing forwarding in only one direction in this manner to also perform packet reception in the other direction. This causes each path to perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably. As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." *Id.* ¶ 25 (emphasis added).

d. "However, because an object of this learn packet is simply for effective path learning to be performed, generating this learn packet with needless frequency actually invites increased line traffic. To prevent such adverse effects, learn-packet generation conditions—that is, a generation frequency, a generation opportunity, and the like—must be appropriately determined." *Id.* ¶ 26.

e. "That is, it is desirable for the learn-packet generation conditions to be a situation wherein the path leading to the destination of the

receive packet includes a trunk, the learn packet being generated for the first time not at the time of the first learning but when the learning result 30 becomes a hit in a destination search. That is, a configuration is desirable that generates the learn packet when a packet is received whose destination is the learned source. Note that at this time, it is desirable to transmit the learn packet to all forwarding devices constituting this trunk. Moreover, it is desirable to also generate the learn packet each subsequent time when the learning result again becomes a hit in the destination search." *Id.* ¶ 28.

f. "To facilitate description, a situation is then supposed wherein the paths of cards #1, #3 are selected as the representative ports of the trunk in the direction wherein the terminal 20 receives 20 and, in contrast, the paths of cards #4, #2 are selected as the representative ports of the same trunk in the direction of transmitting to the terminal 10. In this situation, when no learn packet is generated, as described in conjunction with FIG. 12 to FIG. 14, flooding occurs repeatedly. That is, when a packet of a transmission direction toward the terminal 20 again reaches card #3, because card #3 has not received a packet of a reception direction from the terminal 20,

> no corresponding learning result is had, and flooding is again performed. However, according to this embodiment of the present invention, as above, the learn packet is sent from card #5 to cards #3, #4. As such, at both card #3 and card #4, the MAC address "B" of the terminal 20 is already learned. As such, even when a packet of a transmission direction toward the terminal 20 again reaches card #3, card #3 can specify the path to the terminal 20 according to the learning result provided by the learn packet." *Id.* ¶ 34.

142. It would have been obvious to a POSA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.* Ishimori teaches a learning process where "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33. Ishimori therefore discloses sending a message of the association to each member line card for said plurality of member line cards. It would have been obvious to implement the MAC notification in Smith to perform the method taught in Ishimori to first check whether the MAC address is found in the MAC table, and if not, create a new

record of the association. Both Smith and Ishimori teach that a message of this association is sent to the plurality of member line cards. Ex. 1004 ¶ 63; Ex. 1005 ¶¶ 25, 33. Thus, the combination of Smith and Ishimori renders obvious this limitation.

e. Claim 2: The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC address and the respective ports.

143. In my opinion, the combination of Smith and Ishimori discloses claim 2. Smith teaches that the network device sub-unit 122(2) sends a MAC notification (a message) to update the forwarding engines, but it does not disclose doing so periodically at predefined times. Ex. 1004 ¶ 63. However, Ishimori discloses this. Indeed, Ishimori's disclosures include:

a. "According to the present invention, each packet forwarding device is configured to generate a **learn packet** at a **predetermined timing** under predetermined conditions. That is, even in a situation wherein different paths are selected for a coming direction and a going direction due to trunking or the like and thus only forwarding in one of these directions is performed in a certain device, by generating a learn packet at a **predetermined timing**, this learn packet can force, for example, the device

performing forwarding in only one direction in this manner to also perform packet reception in the other direction. This causes each path to perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably. As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." Ex. 1005 ¶ 25 (emphasis added).

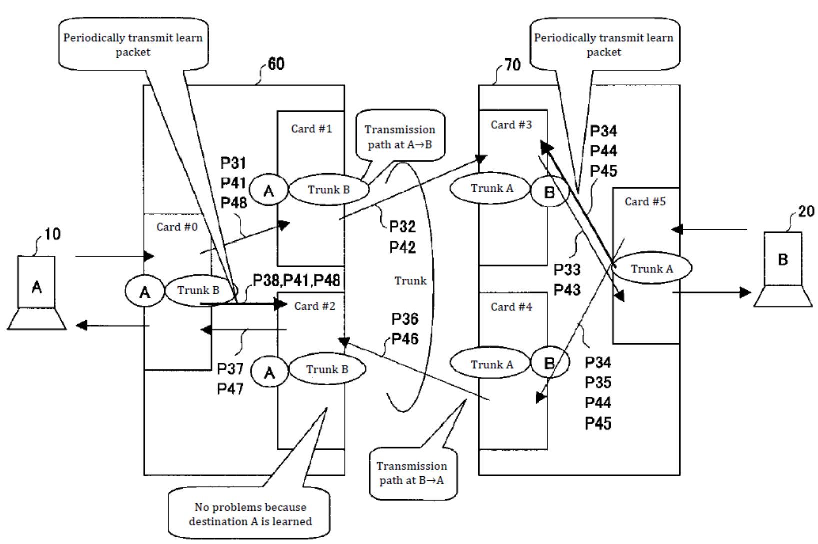
b. "In Figure 15, the path to the destination of the receive packet includes a trunk. Moreover, when, at these nodes, a packet from the same address as the corresponding destination address (DA) is already received and the source-address learning result from this time is had—that is, when there is a hit in the destination search (step S3 in FIG. 2)—a learn packet is transmitted **to all nodes having the corresponding trunk**. That is, in FIG. 15, a packet is first forwarded from the terminal to the terminal 20. When, at each node that this forwarding operation passes through—for example, card #5—a reply packet from the terminal 20 to the terminal 10 is afterward received, at card #5, the learn packet is transmitted to all nodes having the corresponding trunk—that is, cards #3, #4. This causes the path for the corresponding MAC address to be learned at all of these nodes. That is, in this situation, at cards #3, #4, the

path having card #5 is learned for the MAC address "B" of the terminal 20." *Id.* ¶ 33 (emphasis added).

c. "[Problem] An object is to provide, as a packet forwarding system that uses a path learning function applying a link aggregation function, a system that can prevent repeated flooding even when paths differ between a time of transmission and a time of reception. [Resolution Means] A configuration is adopted of periodically sending, according to predetermined conditions, a learn packet to all nodes having a trunk." *Id.*, Abstract.

d. Figure 15 shows "Periodically transmit learn packet":

[FIG. 15]
Block diagram for describing operations of packet forwarding system of embodiment of present invention



Id., Fig. 15.

144. Ishimori teaches that a "learn packet" (message) is generated at a "predetermined timing" (periodically at predefined times). *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33. Therefore, it would have been obvious that this must include at least the second line card. *Id.* Thus, the combination of Smith and Ishimori discloses this limitation.

f. Claim 3: The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

145. In my opinion, the combination of Smith and Ishimori discloses claim 3. Smith discloses that the network device sub-unit 122(2) sends a MAC notification (*e.g.*, the message) to update the forwarding engines. Ex. 1004 ¶ 63. Smith specifies that "[a]fter being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand that this would include the other line cards in virtual network device 202, such as line card 304(4). For example, line card 304(4) (the second line card) would receive the MAC notification (the message) and update its lookup tables. *Id.* ¶ 57. Thus, Smith discloses that in response to the message, the record of association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

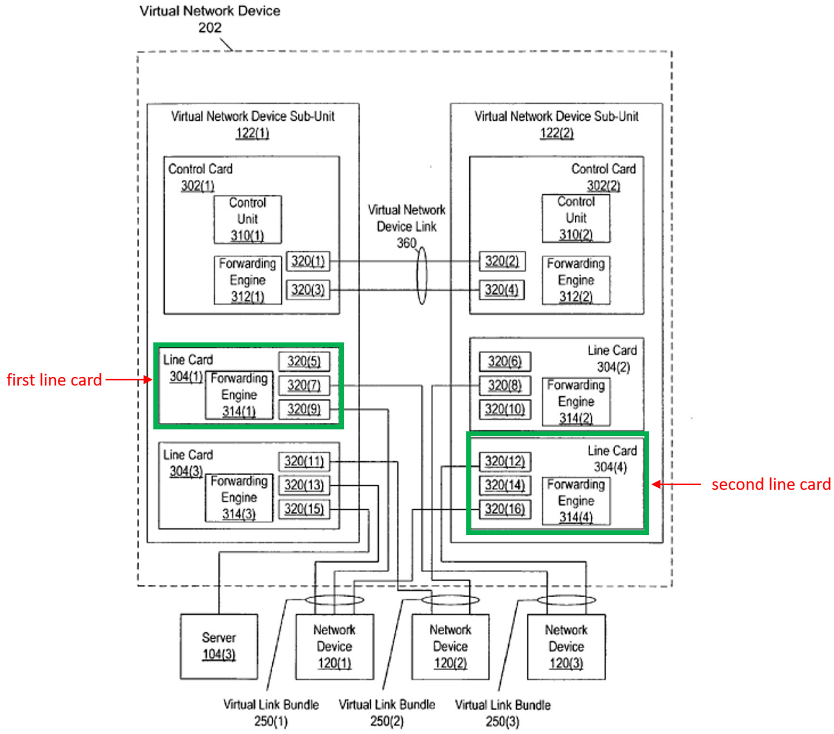


FIG. 3

146. Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of **learning** a source MAC address of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets, an IP packet having an unlearned MAC address as a destination address, a transmission path thereof cannot be specified. As such, in this situation, a broadcast transmission to all nodes (that is,

"flooding") is performed. Then, at this time, path **learning** is carried out by **storing** the source MAC address (SA) had by this IP packet and a port that received such **in a buffer had by each node**." Ex. 1005 ¶ 2 (emphasis added).

b. "The path having this representative port leads to card #3 of the nodes 70. At this card #3 as well, no learning result relating to the terminal 20 that is the destination is had. As such, at card #3 as well, this packet is flooded (step P13). In this manner, this packet reaches the terminal 20 that is the destination through card #5. Note that during the packet forwarding from the terminal 10 to the terminal 20 illustrated in FIG. 9 and FIG. 10, at each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16 (emphasis added).

147. Thus, Ishimori further teaches that the reception path information is learned, which is done by storing the record of association of the MAC address and ingress port to the MAC table. *Id.* ¶ 2. It would have been obvious to do this via the MAC notification as disclosed in Smith. Additionally, during the packet transfer, each of Ishimori's communication cards learns information about the association.

Ishimori discloses that "a learn packet is transmitted to all nodes having the corresponding trunk," which includes at least a second line card. *Id.* ¶ 33. Thus, in Ishimori, each of the communication cards, which includes the second line card, would add a record of the association to the FDB in response to a MAC notification as taught in Smith. The combination of Smith and Ishimori therefore renders obvious this claim.

g. Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

148. In my opinion, the combination of Smith and Ishimori discloses claim 11[c]. Claim 11[c] combines the elements of claims 1[a], 1[b], 1[c], and 11[a]. Therefore, for the same reasons that the combination of Smith and Ishimori discloses claims 1[a], 1[b], 1[c], and 11[a], Smith also discloses claim 11[c]. *See supra* ¶¶ 99–107, 134–136, 127.

h. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress

line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

149. In my opinion, the combination of Smith and Ishimori discloses claim 11[d]. The combination of Smith and Ishimori disclose this limitation for the same reasons that the combination of Smith and Ishimori discloses claims 1[d], 1[e], 1[f], 1[g], and 1[h]. *See supra* ¶¶ 108–120, 137–142.

i. Claim 12: The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.

150. In my opinion, the combination of Smith and Ishimori discloses claim 12. *See supra* ¶¶ 143–144.

j. Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.

151. In my opinion, the combination of Smith and Ishimori discloses claim 13. *See supra* ¶¶ 145–147.

3. Ground 3: Claims 1, 4–7, 10–11, 14–17, and 20 would have been obvious over Smith in view of Ishimori in further view of Edsall

152. I have analyzed claims 1, 4–7, 10–11, 14–17, and 20 and conclude that they would have been obvious over Smith in view of Ishimori in further view of Edsall. I provide a detailed analysis of each claim limitation below.

a. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

153. In my opinion, to the extent this is not disclosed by the combination of Smith and Ishimori, it is disclosed in further view of Edsall. Edsall teaches that the forwarding engine learns the source MAC address of a frame received at the ingress card for the first time. "If the frame is received at the ingress card for the first time, the forwarding engine also 'learns' a source MAC address of the frame." Ex. 1006 at 18:39–41. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 18:42–44. "[I]f there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* at 18:54–55.

154. It would have been obvious to use Edsall's learning methods with the Smith-Ishimori MAC tables, and there would have been a reasonable expectation of success in doing so.

b. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.

155. In my opinion, to the extent the combination of Smith and Ishimori does not render obvious this limitation, it would have been obvious to combine Smith and Ishimori with Edsall. Edsall teaches that the forwarding engine learns the source MAC address of a frame received at the ingress card for the first time. *Id.* at 18:39–41. Edsall further teaches that "if there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* at 18:54–55. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 18:42–44. It then floods copies of the fabric frame to all the egress line cards of the network switch, which Edsall calls the "flood-to-fabric (FF) operation." *Id.* at 18:47–50. This "forces each forwarding engine associated with each egress card to either (i) update its current L2 forwarding table entry with the newly-learned source MAC address and index of the frame or, if there is not a current entry, (ii) learn the source address/index of the frame." *Id.* at 18:50–55.

156. It would have been obvious to add Edsall's learning method and flood-to-fabric operation in the Smith-Ishimori path learning operations. A POSA would

have been able to implement this teaching with a reasonable expectation of success, because the combination of Smith and Ishimori already teach that a message of new associations are sent to the plurality of member line cards. Ex. 1004 ¶ 63; Ex. 1005 ¶ 16. It would have been within the knowledge of a POSA to create new entries in a FDB when an association of a MAC address and port does not yet exist. Thus, the combination of Smith, Ishimori, and Edsall renders obvious this limitation.

c. Claim 4: The method according to claim 3 and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

157. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 4. Edsall discloses that the "PI indicator" is asserted when a forwarding table entry for the MAC address is learned through one of the ports (*e.g.*, data packets transmitted via a port of the line card) "as opposed to through the switch fabric" (*e.g.*, received from another of the line cards). Ex. 1006 at 6:46–64. Thus, the PI indicator is different for the MAC address learned through one of the ports, which correspond to the first type of records in the claim ("data packets transmitted via a port of the line card") "as opposed to through the switch fabric, which corresponds to the second type of records ("messages from another of the line cards"). Ex. 1006 at 6:46–64.

158. Additionally, during prosecution, the Examiner found that Edsall disclosed this limitation and the applicant did not amend the claims based on this rejection. Ex. 1002 at 82, 120–122. The Examiner determined that Edsall also discloses that the "PI indicator is asserted for a destination MAC address entry of the forwarding table on the egress card and the DI contained in the switched fabric frame (*i.e.*, the ingress DI) is different from the DI stored in this egress forwarding table (*i.e.*, the egress DI)." Ex. 1006 at 18:56–19:5; *see also* Ex. 1002 at 82. Based on my review of the file history, it is my understanding that this disclosure from Edsall was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 120–122.

d. Claim 5[a]: The method according to claim 4, and comprising: associating a respective aging time with each of the records;

159. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 5[a]. Ishimori's disclosures include:

a. "Furthermore, in this system, based on an approach of effectively utilizing a limited buffer storage capacity, upon learning, when there is no DA search hit for the same address in a certain period, the learning result relating to this MAC address is deleted. This operation leading to deletion is referred to as "**aging.**" Specifically, as illustrated in FIG. 5, the buffers of the cards of each node have **a**

"hit bit" corresponding to the learned MAC address; when this is learned, the hit bit = "1". Then, at the **first aging process, this hit bit is cleared** (that is, made to be "0"; steps S11, S12). Here, the learning result is not yet deleted. Then, **when the hit bit is still "0" at the next aging process** (step S14, step S15, "No" at step S11), the **corresponding MAC address is deleted from the buffer** (step S13). Meanwhile, when overwriting learning is performed previous to this or when there is a hit at the DA search (that is, the destination search operation of step S3 in FIG. 2 above), the **hit bit is again set to "1"** (as a result, step S11 "Yes" and step S12). This configuration prevents an address currently being used (that is, being used for communication) from being deleted from the buffer." Ex. 1005 ¶ 9 (emphasis added).

b. "This aging process is further described in conjunction with FIGS. 6 to 8. Steps P1 to P6 in FIG. 7 are the same steps as steps P1 to P6 in FIG. 4 above. Now, each time an IP packet is forwarded, at steps P1, P2, the MAC address "A" is learned at card #0, and the hit bit thereof is made to be "1". Likewise, at steps P2, P3, the MAC address "A" is learned at card #1 and card #2. Continuing in the

same vein, at steps P4, P5 and steps P5, P6, the MAC address "B" is learned at card #2 and card #0." *Id.* ¶ 10 (emphasis added).

c. "Afterward, as illustrated in the uppermost row in FIG. 8, the first aging process clears the respective hit bits of the MAC addresses "A", "B" at each card #0, #1, #2 (steps S12). It is then supposed that before the second aging process, the next IP packet, likewise addressed to terminal B 20, is sent from terminal A 10 (step P7). At card #0 that receives this, the source MAC address "A" is learned by overwriting. At the same time, in the destination search, the destination MAC address "B" becomes the search target. As such, the MAC addresses "A", "B", which are temporarily cleared to "0" as above, have their hit bits respectively returned to "1"." *Id.* ¶ 11 (emphasis added).

d. "Then, this IP packet is forwarded to card #2 according to the learning result regarding the MAC address "B" that yielded a hit in the destination search (step P8). Then, in card #2 as well, theMAC addresses "A", "B" temporarily cleared to "0" as above have their hit bits respectively returned to "1" by the same operation as above. It is then supposed that afterward, the next aging-process timing arrives without either of the IP packets of terminals A, B 10,

20 being received. Here, as illustrated in the lowermost row in FIG. 8, at card #0 and card #2, as above, the hit bits of the MAC addresses "A", "B" are each returned to "1" at steps P7, P8. As such, these are again cleared to "0". However, the corresponding learning result is not yet erased at this point (step S12). Meanwhile, for card #1, no corresponding IP packet passes through during this time, and the hit bits of the MAC addresses "A", "B" remain "0". As such, the corresponding learning result is deleted at this point (step S13)." *Id.* ¶ 12 (emphasis added).

160. Ishimori teaches that "the buffers of the cards of each node have a 'hit bit' corresponding to the learned MAC address; when this is learned, the hit bit = '1'." *Id.* ¶ 9. Therefore, Ishimori associates a respective aging time with each of the records. "[A]t the first aging process, this hit bit is cleared." *Id.* "[W]hen the hit bit is still '0' at the next aging process," "the corresponding MAC address is deleted from the buffer." *Id.* Thus, the hit bit is how Ishimori associates a respective aging time with each of the records.

161. It would have been obvious to implement a "hit bit" on the lookup tables in the line cards of Smith and Edsall (where Edsall leverages the PI Indicator) in order to associate a respective aging time with each of the records.

e. Claim 5[b]: refreshing the records in the FDB responsively to further packets transmitted by the line cards; and

162. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 5[b]. As described above, Ishimori teaches that when the hit bit is "0," "the corresponding MAC address is deleted from the buffer." *Id.* ¶¶ 9, 11. Ishimori then teaches that when the destination MAC address is received at card #0, the source MAC address is learned by "overwriting," and the hit bit is "returned to '1'" (*e.g.*, refreshing the records in the FDB responsively to further packets transmitted by the line cards). *Id.* ¶ 11.

163. It would have been obvious to a POSA implement a "hit bit" on the lookup tables in the line cards of Smith and Edsall (where Edsall leverages the PI Indicator) and to refresh the hit bit by learning the source MAC address of the data packets transmitted by the line cards.

f. Claim 5[c]: removing the records from the respective FDB if the records are not refreshed within the respective aging time.

164. In my opinion, the combination of Smith and Ishimori discloses claim 5[c]. Ishimori teaches that if the hit bit remains "0" in the next aging process, then the corresponding MAC address is deleted. *Id.* ¶¶ 9, 12. Thus, Ishimori discloses removing the records from the FDB if the records are not refreshed within the respective aging time.

165. It would have been obvious to a POSA implement a "hit bit" on the lookup tables in the line cards of Smith and Edsall (where Edsall leverages the PI Indicator) to remove the records from the respective lookup tables in Smith-Edsall if the record has not been refreshed by the next aging process as taught by Ishimori.

g. Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via switching core of the network node to at least the second line card.

166. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 6. As I explained above, Smith discloses this limitation. *See supra* ¶¶ 123–125. Edsall further discloses the "plurality of line cards" are "interconnected by a switch fabric 550." Ex. 1006, 8:20–27. The "switch fabric" in Edsall could be a "switching core." Thus, a POSA would have understood that the packet is sent from the first line card via a switching core to at least the second line card.

h. Claim 7: The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.

167. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 7. Smith teaches that its virtual network device includes several line cards, which include several interfaces. Ex. 1004 ¶ 46. "[W]hen updating control protocol

behavior of virtual link bundle 250(1), a user can simply access virtual network device sub-unit 122(1) (instead of accessing both virtual network device sub-units 122(1) and 122(2)). *Id.* Virtual network device sub-unit 122(1) can then automatically propagate to network device 122(2) any changes made by the user to the control protocols." *Id.* ¶ 59. Smith teaches that "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶ 62.

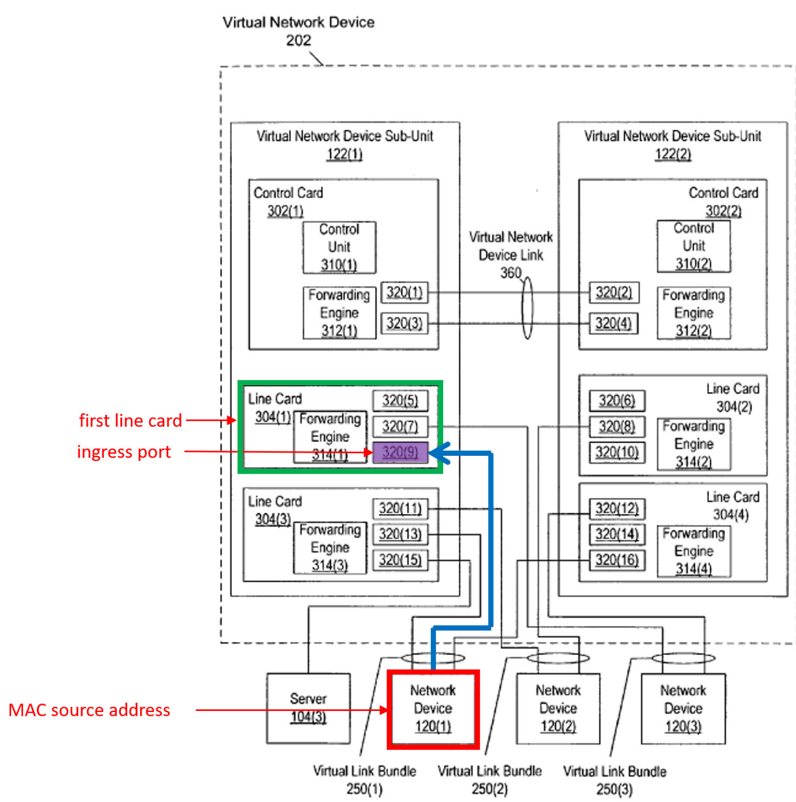


FIG. 3

168. Thus, if the record in the FDB of the first line card 304(1) associates the MAC source address (the MAC address of virtual device 120(1)) with a port

different from the one of the ports on which the data packet was received (a port other than 320(9)), the MAC notification frames will notify and update the L2 tables in the virtual network device 202, which would include at least the second line card 304(4) to indicate the record has been changed. Similarly, Ishimori teaches that the routes are transmitted via a "learn packet." Ex. 1005 ¶ 25. A POSA would understand that the "learn packet" as taught in Ishimori functions as a synchronization packet. During the packet transfer, each of the communication cards learns information about the route (*e.g.*, the second line card). *Id.* ¶¶ 28, 33.

169. Edsall further discloses that the forwarding engine generates an MN frame (*e.g.*, synchronization packet) that may get sent to the SMC (switch management card) to ensure that FwdT0 (*e.g.*, the forwarding table) is synchronized. Ex. 1006 at 17:26–38. The forwarding engine also asserts an appropriate bit of the POE field (port-of-exit field) when generating the MN frame, which is a port different from one of the ports on which data was received. *Id.* It would have been obvious to a POSA to implement the MN frame from Smith to include the POE field in order to note the port interface of the switch fabric. *Id.* at 6:24–25. This disclosure was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83, 120–122. Thus, the combination of Smith, Ishimori, and Edsall discloses this limitation.

i. Claim 10[a]: The method according to claim 1, and comprising: conveying a further data packet, received

from a further MAC source address, to the second line card for transmission over the network;

170. In my opinion, Smith discloses claim 10[a]. *See also supra* ¶¶ 108–111. Smith teaches that the uplink interface receives a data packet with the "sending device's MAC address," which is a MAC source address. Ex. 1004 ¶ 54. Based on Figure 3, a further data packet received from a further MAC source address (such as network device 120(3)) would be conveyed to the second line card 304(4) for transmission over the network.

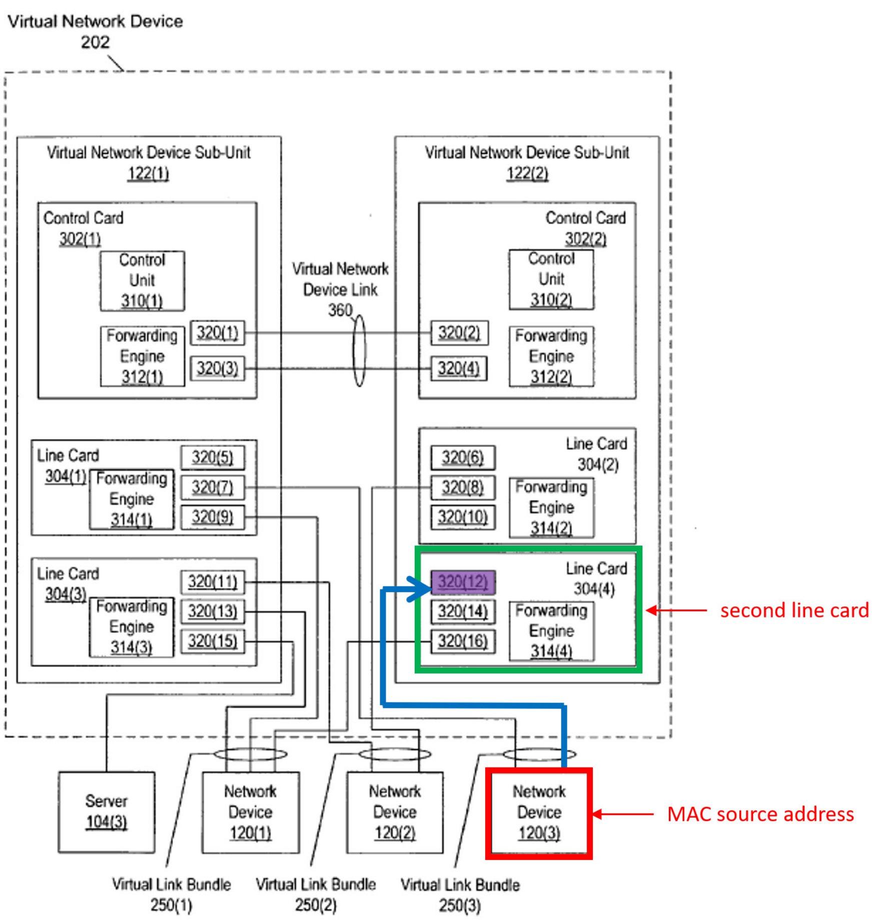


FIG. 3

171. To the extent Smith does not render obvious this claim, Edsall does. Edsall teaches that in each subsequent frame, the encoded address recognition logic (EARL) circuit looks up the MAC address and "sends the corresponding rewrite information over the local bus after the frame" (*e.g.*, conveying a further data packet to the second line card for transmission over the network). Ex. 1006 at 14:22–34. This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122.

j. Claim 10[b]: checking the further MAC source address against the records in the FDB of the second line card; and

172. In my opinion, Smith discloses claim 10[b]. *See also supra* ¶¶ 117–118. Smith teaches that that the virtual network device sub-unit learns the source identifier of the sending device, which is the MAC source address. Ex. 1004 ¶ 65. These identifiers are stored in a lookup table (*e.g.*, FDB) on virtual network device sub-unit. *Id.* ¶ 61. The MAC source address, which is the MAC address of network device 120(3), would be checked in the records of the FDB of said second line card 304(4).

173. To the extent Smith does not disclose this limitation, Edsall does. Edsall teaches checking to see if the rewrite information matches, in other words, checking the MAC source address against the records in the FDB of the second line card). Ex. 1006 at 14:22–34 ("The destination port circuitry (or, alternatively, a

UDlink or central rewrite engine) matches the frame with the rewrite information and modifies the frame as needed by replacing, *inter alia*, the destination and source MAC addresses.”). This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122.

k. Claim 10[c] responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.

174. In my opinion, the combination of Smith, Ishimori, and Edsall renders obvious claim 10[c]. *See also supra* ¶¶ 117–118. Smith teaches that that the network device sub-unit 122(2) sends a MAC notification (*e.g.*, a message) to update the forwarding engines (*e.g.*, sending a further message to inform at least the first line card) when it learns of a new association (*e.g.*, the further record). Ex. 1004 ¶ 63. “After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address.” *Id.* A POSA would understand that the MAC notification (*e.g.*, message) is therefore sent to inform at least the first member line card (*e.g.*, line card 304(1)) of the new association.

175. Edsall further teaches that the forwarding engine “modifies the frame as needed by replacing . . . the destination and source MAC addresses” (*e.g.*, adding

a further record with respect to the MAC source address to the FDB). Ex. 1006 at 14:22–34. This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122. It would have been obvious to use any of the learning methods from Smith, Ishimori, or Edsall, which all disclose updating a second line card with a further record and sending a message to inform at least the first line card of the further record.

l. Claim 11[a]: a switching core;

176. To the extent this is not disclosed by the combination of Smith and Ishimori, Edsall discloses this. The ’400 patent describes the “switching core” as linking the multiple line cards. Ex. 1001 at 6:8–10. Edsall teaches that the “plurality of line cards . . . are interconnected by a switch fabric 550.” Ex. 1006 at 8:20–27. Thus, the “switching fabric” in Edsall could be a “switching core.”

m. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address

does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

177. In my opinion, the combination of Smith, Ishimori, and Edsall renders obvious claim 11[d]. This limitation is a combination of claims 1[d], 1[e], 1[f], 1[g], and 1[h]. Therefore, the combination of Smith, Ishimori, Edsall renders obvious this limitation for the same reasons that the combination of Smith, Ishimori, and Edsall renders obvious claims 1[d], 1[e], 1[f], 1[g], and 1[h]. *See supra* ¶¶ 108–120, 137–142, 153–156.

n. Claim 14: The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

178. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 14. *See supra* ¶¶ 157–158.

o. Claim 15: The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.

179. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 15. *See supra* ¶¶ 159–165.

p. Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

180. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 16. *See supra* ¶ 166.

q. Claim 17: The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.

181. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 17. *See supra* ¶¶ 167–169.

r. Claim 20: The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

182. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 20. *See supra* ¶¶ 170–175.

4. Ground 4: Claims 8–9 and 18–19 would have been obvious over Smith in view of Ishimori in further view of Zelig

183. I have analyzed claims 8–9 and 18–19 and conclude that they would have been obvious over Smith in view of Ishimori in further view of Zelig. I provide a detailed analysis of each claim limitation below.

a. Claim 8: The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

184. In my opinion, the combination of Smith, Ishimori, and Zelig discloses claim 8. Zelig's disclosures include:

a. "In another preferred embodiment, at least one of the backup virtual bridges is connected by a single secondary virtual connection to a selected one of the primary virtual bridges, such that upon the failure of the at least one of the primary virtual bridges with which the at least one of the backup virtual bridges is associated, the at least one of the backup virtual bridges transmits and receives the data packets over the network via the selected one of the primary virtual bridges over the single secondary virtual connection. Preferably, **each of the primary virtual bridges** is

adapted to **maintain a respective media access control (MAC) table**, and to **forward the data packets in accordance with entries in the MAC table**, and wherein each of the primary virtual bridges is adapted, upon detecting the failure of the at least one of the primary virtual bridges, to update the entries in the respective MAC table that point to the at least one of the primary virtual bridges so as to point instead to the selected one of the primary virtual bridges." Ex. 1007 ¶ 31 (emphasis added).

b. "FIG. 1 is a block diagram that schematically illustrates a **VPN** 20 with a hierarchical **VPLS** topology, implementing a protection Scheme in accordance with a preferred embodiment of the present invention. **VPN is built around a virtual private LAN service (VPLS)**, operating within a network 22, typically an IP or MPLS network. **The VPLS is based on virtual bridges** 30, 32,34, 36,38, and 40,or VPLS-capable PEs, which are connected by PWs 70, 72, 74 and 76 through network 22. Although for clarity of illustration, network 22 includes only a small number of PES and represents only a single VPLS instance, the principles embodied in this network may be extended in a straightforward manner to larger

networks and to multiple VPLS instances." *Id.* ¶ 42 (emphasis added).

c. "Three primary virtual bridges 30, 32 and 34, referred to as primary core nodes, are connected with each other in a full mesh with PW connections 70. Typically, the PW connections comprise MPLS tunnels, but they may alternatively comprise virtual connections of other types, such as GRE or L2TP tunnels. Each of the primary core nodes 30, 32 and 34, is paired with a corresponding backup virtual bridge, referred to as a standby core node, 36, 38 and 40, respectively. The standby core nodes are connected in the network by redundant backup connections 72. Each standby core node has a topology identical to its corresponding primary core node. For example, standby core node 36 has the same topology image as primary core node 30. Each of the primary and standby core nodes is connected to all the other core nodes in the network except for the standby or primary core node with which it is paired. An optional connection 79 between a primary core node and its corresponding standby core node may also be included, as described hereinbelow." *Id.* ¶ 43.

185. Zelig discloses that its VPN 20 contains multiple primary virtual bridges 30, 32, and 34. *Id.* ¶¶ 42–43. It would have been obvious to a POSA that the virtual bridges servicing VPN 20 are configured to serve that VPN. Zelig further teaches that "each of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table . . . ." *Id.* ¶ 31. Thus, Zelig discloses that each MAC bridge maintains its own MAC table, and because the MAC bridge serves only that VPN instance, the records associating the MAC addresses with the respective ports are maintained independently for each VPN instance.

b. Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

186. In my opinion, the combination of Smith, Ishimori, and Zelig discloses claim 9. Smith teaches that the MAC notification is "distributed to any other forwarding engines within virtual network device sub-unit 122(2)." Ex. 1004 ¶ 63. Zelig provides that "VPN 20 is built around a virtual private LAN service (VPLS), operating within a network 22." Ex. 1007 ¶ 42.

187. It would have been obvious to a POSA to adopt the notification scheme in Smith to the architecture in Zelig so that the MAC notification, which is a message, is sent to all the line cards that serve the VPLS instances. Furthermore, the

IEEE 802.1Q standards, which is cited by Zelig, Ex. 1007 ¶ 2, disclose a VLAN identifier (VID) as a twelve-bit field that "uniquely identif[ies] the VLAN to which the frame belongs." Ex. 1008 § 9.3.2.3; *see also* Ex. 1016 § 9.6. Thus, it would have been within the knowledge of a POSA to identify the VPLS instance in the message, such as using the VID, in order to inform all the line cards in the VPLS.

c. Claim 18: The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

188. In my opinion, the combination of Smith, Ishimori, and Zelig discloses claim 18. *See supra* ¶¶ 184–185.

d. Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

189. In my opinion, the combination of Smith, Ishimori, and Zelig discloses claim 19. *See supra* ¶¶ 186–187.

5. Ground 5: Claims 9 and 19 would have been obvious over Smith in view of Ishimori, Zelig, and 802.1Q-1998

190. I have analyzed claims 9 and 19 and conclude that they would have been obvious over the combination of Smith, Ishimori, Zelig, and 802.1Q-1998. I provide a detailed analysis of each claim limitation below.

a. Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

191. In my opinion, the combination of Smith, Ishimori, Zelig, and 802.1Q-1998 discloses claim 9. *See supra* ¶¶ 186–187.

b. Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

192. In my opinion, the combination of Smith, Ishimori, Zelig, and 802.1Q-1998 discloses claim 19. *See supra* ¶ 189.

IX. ADDITIONAL REMARKS

193. I currently hold the opinions expressed in this Declaration. I reserve the right to further explain and supplement my opinions as I may acquire additional information and/or attain supplemental insights that may result in added observations.

194. I hereby declare that to the best of my knowledge all statements made are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of this proceeding.

Executed this 23rd day of December, 2022.

Dr. Nicholas Bambos

Appendix A

CURRICULUM VITAE

Nicholas Bambos

R. Weiland Professor of Electrical Engineering
and of Management Science & Engineering

Stanford University

December 2022

Nick Bambos is a R. Weiland Professor in the School of Engineering at Stanford University, having a joint appointment in the Department of Electrical Engineering and the Department of Management Science & Engineering. He has also been the Fortinet Founders Department Chairman of the Management Science & Engineering Department (2016-20). He heads the Computer Systems & Networks Performance Engineering Lab at Stanford. This is comprised of PhD students and industry visitors engaged in research projects, which require solid understanding of the physical principles of computing and networking hardware and software architectures (e.g., digital design, file systems), but largely focus on system architectures and high-performance engineering of computer systems and networks. His research contributions span the areas of wireless & wireline networking, the Internet, cloud computing and data centers, high-speed networking and packet switching, multimedia streaming, information service engineering, computer systems security, etc. His methodological interests and contributions span the areas of stochastic modeling and control, online task scheduling, routing and distributed processing, network control, etc.

He has graduated over 40 Ph.D. students (including two postdocs), who have moved on to key positions in academia (Stanford, CalTech, Michigan, GaTech, NYU, USC, Columbia, etc.), the information technology industry (Cisco, IBM, Broadcom, Qualcomm, Nokia, MITRE, ST Micro, Intel, Samsung, Google, Facebook, Twitter, etc.), technology startups and the finance tech industry. From 1999 to 2005 he served as Director of the Stanford Networking Research Center project, a partnership between Stanford and information technology industries, involving tens of corporate members, faculty and doctoral students.

He received his Ph.D. in Electrical Engineering and Computer Sciences (EECS) from the University of California at Berkeley (1989), as well as the M.S. in EECS (1987) and the M.A. in Mathematics (1989) from the same University. He graduated in Electrical Engineering from the National Technical University of Athens-Greece (1984) with first class honors. Before joining Stanford as an Associate Professor in 1996, he served as Assistant (1990-95) and tenured Associate Professor (1995-96) in the Electrical Engineering Department of the University of California at Los Angeles (UCLA).

Nick Bambos has received several best research papers awards, has held the Cisco Systems Faculty Development Chair (1999-2003) in computer networking at Stanford, and has won the IBM Faculty Award (2002) for high-impact research in performance engineering of computer systems and networks, as well as the Griffin Award (1997). He has been the David Morgenthaler Faculty Scholar (1996-99) at Stanford, and has received the National Young Investigator Award (1992) from the National Science Foundation (NSF) for research in computer networks and distributed computing architectures, as well as the NSF Research Initiation Award (1990) for studies in performance modeling of computer systems. He has also been a U.C. Regents Fellow, a David Gale Fellow, and an Earl Anthony Fellow at U.C. Berkeley.

He has served as editor of various research journals, on international technical and scientific committees, and on review panels for networking and computing technologies. He has also served on the technical boards of start-up companies, as a consultant for high-technology development and management, and as an expert witness in high-profile patent litigation cases involving networking, computing and information technologies.

EDUCATION

- **Ph.D. in Electrical Engineering and Computer Sciences**,
  University of California at Berkeley, 1989.

- **M.A. in Mathematics**,
  University of California at Berkeley, 1989.

- **M.S. in Electrical Engineering and Computer Sciences**,
  University of California at Berkeley, 1987.

- **Diploma in Electrical Engineering -** First Class Honors,
  National Technical University of Athens-Greece, 1984.

PROFESSIONAL EXPERIENCE

- **R. Weiland Professor** in the School of Engineering, Stanford University, 2016 – present.

- **Fortinet Founders Department Chairman,** Department of Management Science & Engineering, Stanford University, 2016 – 2020.

- **Professor**, Department of Management Science & Engineering, and Department of Electrical Engineering, Stanford University, 2003 – present.

- **Head,** Computer Network Architecture and Performance Engineering Research Lab (NetLab), Stanford University, 1998 – present.

- **Director**, Stanford Networking Research Center (SNRC) Project Initiative, Stanford University, 1999 – 2005.

- **Associate Professor**, Department of Management Science & Engineering, and Department of Electrical Engineering, Stanford University, 1999 – 2003.

- **Associate Professor**, Department of Engineering-Economic Systems & Operations Research, and (courtesy appointment at) Department of Electrical Engineering, Stanford University, 1996 – 1999.

- **Associate Professor** (tenured), Department of Electrical Engineering, University of California at Los Angeles, 1995 – 1996.

- **Assistant Professor**, Department of Electrical Engineering, University of California at Los Angeles, 1989 – 1995.

HONORS AND AWARDS

- **2022 Best Paper Award for Telecommunications and Network Analytics** from INFORMS on for the paper "Robust Power Management via Learning and Game Design" (Z. Zhou, P. Mertikopoulos, A. Moustakas, N. Bambos, P. Glynn), INFORMS Annual Meeting, IN, October 2022.

- **Outstanding Paper Award at Healthcom** for the paper "Physiological Waveform Imputation of Missing Data using Convolutional Autoencoders" (with D. Miller, A. Ward, D. Scheinker, Andrew Shin). *Proceedings of the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom'18), Ostrava, Czech Republic, September 2018.*

- **Best Paper Award in Green Communication and Computing** by the TAOS (Transmission, Access & Optical Systems) Technical Committee for the paper: *Dynamic Resource Management in Virtualized Data Centers with Bursty Traffic* (with M. Valdez-Vivas, J. Apostolopoulos) presented in the IEEE International Communications Conference (ICC'14) in Sydney, Australia, June 2014.

- **"Best of Globecom" Paper Award** (top 2%) for the paper: *Dynamic Resource Management in Cloud Computing with Frictions and Congestion "Weather"* (with M. Valdez-Vivas, J. Apostolopoulos) presented in IEEE Global Communications Conference (Globecom'14) in Austin, TX, December 2014.

- **2013-14 Eugene L. Grant Teaching Award**, Stanford University.

- **Best Journal Paper Award for 2011** from the IEEE Multimedia Communications Technical Committee for the paper: *Channel, Deadline, and Distortion Aware Scheduling of Video Streams over Wireless Links* (with A. Dua, C. Chan, J. Apostolopoulos), IEEE Trans. on Wireless Communications, 9(3):1001-1011, 2010.

- **HP Faculty Award**, Hewlett-Packard Laboratories, Palo Alto, 2006

- **Cisco Systems Faculty Scholar**, School of Engineering, Stanford University, 1999 - 2003.

- **IBM Faculty Award**, IBM Corporation Research Labs, 2002.

- **Dana Adams Griffin Award**, School of Engineering, Stanford University, 1997.

- **David Morgenthaler Faculty Scholar**, School of Engineering, Stanford University, 1996 - 1999.

- **National Young Investigator Award**, National Science Foundation, 1992 - 1997.

- **Research Initiation Award**, National Science Foundation, 1990 - 1992.

- **EECS Departmental Scholar**, University of California at Berkeley, 1988 - 1989.

- **U.C. Regents Fellow**, University of California at Berkeley, 1986 - 1988.

- **Earl C. Anthony Fellow**, University of California at Berkeley, 1985 - 1986.

- **David and Sylvia Gale Fellow**, University of California at Berkeley, 1984 - 1985.

- **First Class Honors** for top graduating students in 1984 (among 150 students), Electrical Engineering Department, National Technical University of Athens-Greece, 1984.

- **Academic Excellence Awards** (annual), Hellenic State Scholarships Foundation, Greece, 1979 - 1984.
- **Academic Prizes** (annual), Hellenic Engineering Society, Greece, 1979 - 1984.
- **First Class Honors** for top students admitted to the University in 1979 (among tens of thousands of students participating in a national-level entrance exam), National Technical University of Athens-Greece, 1979.

DISTINGUSHED INVITED LECTURES AND TALKS (selected; additional talks on pg. 19)

- *Keynote Talk – Geometry of Resource Scheduling in Service Systems.* The 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, July 21, 2018.
- *Plenary Talk -- Transmitter Power Control in Wireless Networking***.** The 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, Hawaii, March 6, 2018.
- *Keynote Address – Power Control and Bandwidth Management in Wireless Networking,* IEEE Wireless On-Demand Network Systems and Services Conference (WONS), Jackson Hole, WY, February 2017.
- *Plenary Address – Digital Living 2030,* Shandong Chinese Academy of Sciences, Jinan, China, January 2017.
- *Keynote Address – A Risk Management View to Information Security*, IEEE Conference on Decision and Game Theory for Security, Berlin, Germany, Nov. 2010.
- *Keynote Address – Wireless Computing: From Infrastructure to Services*, IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Athens, September 2007.
- *Invited Distinguished Lecture* - *Three Control Problems in Packet Switching*, Workshop on Mathematics of Networks, Ecole Normale Superieure, Paris, France, June 2007.
- *Invited Distinguished Talk - Throughput and Power Issues in Packet Switches*, Center for Ultra-Broadband Information Networks, University of Melbourne, Australia, August 2007.
- *Keynote Talk* **-** *Opportunities and Challenges in Wireless Computing and Multimedia Networking*, Network Ventures Conference, Redwood Shores, CA, March 2005.
- *IBM Distinguished Talk* **-** *Resource Management and Load Balancing in Flexible Computing/Networking Service Platforms*, IBM Research Labs, Almaden, CA, June 2005
- *Distinguished Speaker* - *Power Control in Wireless Networks,* Wireless Industry Forum, Electrical Engineering Department, University of Texas at Austin, April 2004.
- *Plenary Talk* - *Power Control in Wireless Networks* and member of Panel of Experts on Network Technology Tends, IEEE Workshop on Local and Metropolitan Area Networks (LANMAN), pp. 69-74, CA, April 2004
- *Invited Lecture* **-** *Geometry of Packet Scheduling in Communication Switches,* Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Nov. 2004.
- *Plenary Talk – Next Generation Wireless System Architectures* and member of Panel of Experts on Wireless Technologies in the IEEE International Conference in Broadband Networks (BroadNets), San

Jose, CA, October 2004.

- *Plenary Talk – Next Generation Wireless Data Networks: Control Plane Principles and Architectures*. Strategic Architecture Council of Sun Microsystems, Nov. 2003

- *International Expert* invited to serve on the International Evaluation Committee for network engineering research at INRIA (the French/European National Institute of Informatics and Automation of 900 scientists working in six campuses), 2003-2004.

UNIVERSITY AND PROFESSIONAL SERVICE

- *Academic Senate* of Stanford University (Member), 2005 – 2010

- *Committee on Research*, Stanford University (Member), 2009 – 2012.

- *Member of the Steering Board*, Conference on Decision and Game Theory for Security (GameSec), 2010-present.

- *Member of the Scientific Board*, LINCS – Laboratory on Information, Networking and Communication Sciences, Paris, France (founded by INRIA, Institut Telecom, France Telecom and Alcatel-Lucent), 2010-present.

- *Member the Editorial Boards* of the Wireless Networks Journal (2001-07), Computer Networks Journal (2000-02), Queueing Systems Theory and Applications (1999-2007).

- *Member of the Technical Program Committees* of the various networking/computing conferences (IEEE Infocom 2001-03, IEEE Globecom 2001-03, IEEE Mobicom 2002, IEEE VTC 2003, IEEE MobiWac 2005, SpasWin 2006, BroadNets 2006-07, Wireless Broadband World Forum 2006-07, GameComm 2008)

- *Technical Reviewer* for several research journals in computing and networking, as well as in queueing systems and applied probability (IEEE/ACM Transactions on Networking, IEEE Tran. on Vehicular Technology, IEEE Tran. on Automatic Control, Advances in Applied Probability, Journal of Applied Probability, Annals of Applied Probability, Operations Research, Journal of the ACM, Math of OR, IEEE Trans. on Communications)

- *Member of various Technical Panels* of the National Science Foundation (NSF) and other research and technology development agencies (NSF Information Technology Research Panel, 2002; INRIA Project Evaluation Panel, 2003; NSF CAREER Panel, 2005; NSF Foundations Panel, 2006; NSF NeTS CAREER Panel, 2006; NSF FIND Panel, 2007; NSF CDI Panel, 2008).

RESEARCH PUBLICATIONS, REPORTS, TALKS, ETC.

BOOK CHAPTERS:

1. **Power Control in Wireless Networking: Basic Principles and Core Algorithms** (with S. Gitzenis). Signal Processing for Mobile Communications Handbook (M.Ibnkahla, Editor), CRC Press, 2004.

2. **Transmitter Power Control in Wireless Computing** (with S. Gitzenis). Handbook of Mobile Computing (I. Mahgoub, Editor), CRC Press, 2004.

3. **Adaptive Batch Scheduling for Packet Switching with Delays** (with K. Ross). High-Performance Packet Switching Architectures (Editors: I. Elhanany, M. Hamdi), Springer-Verlag, pp. 66-80, 2006.

4. **Geometry of Packet Scheduling: Maximal Throughput Cone Scheduling Algorithms** (with K. Ross). High-Performance Packet Switching Architectures (Editors: I. Elhanany, M. Hamdi), Springer-Verlag, pp. 82-100, 2006.

5. **Autonomic Admission Control for Networked Information Servers** (with G. Paleologo). Telecommunications Network Design and Management, Operations Research and Computer Science Interfaces, Kluwer, vol. 23, pp. 227-244, 2003

JOURNAL PAPERS:

6. **Smart Greedy Distributed Energy Allocation: a Random Games Approach** (I. Bistritz, A. Ward, Z. Zhou, N. Bambos). *IEEE Transactions on Automatic Control* (accepted*)*.

7. **No Weight-Regret Learning in Adversarial Bandits with Delays** (I. Bistritz, Z. Zhou, X. Chen, N. Bambos). *Journal of Machine Learning Research (JMLR)* (accepted).

8. **Distributed Stochastic Optimization with Large Delays** (Z. Zhou, P. Mertikopoulos, N. Bambos, P. Glynn and Y. Ye). *Mathematics of Operations Research*, accepted and published online in Dec. 2021 (forthcoming as regular publication).

9. **Wireless Body-Area Network Control Policies for Energy-Efficient Health Monitoring** (Y.B. David, T. Geller, I. Bistritz, I. Ben-Gal, N. Bambos, E. Khmelnitsky). Sensors Journal, 21/4245, pp. 1-21, June 2021.

10. **Prediction of Prolonged Opioid Use After Surgery in Adolescents: Insights From Machine Learning** (*A. Ward*, T. Jani, E. De Souza, D. Scheinker, N. Bambos, A. Anderson). *Journal of Anesthesia & Analgesia,* 133(2):304-313, 2021.

11. **Consensus-Based Stochastic Control for Model-Free Battery Cell Balancing** (I. Bistritz, N. Bambos). *IEEE Transactions on Control of Network Systems*, 8(3):1139-1150, Sept. 2021.

12. **Robust Power Management via Learning and Game Design** (Z. Zhou, P. Mertikopoulos, A. Moustakas, N. Bambos, P. Glynn). *Operations Research*, 69(1):331-345, 2021.

13. **One for All and All for One: Distributed Learning of Fair Allocations with Multi-Player Bandits** (I. Bistritz, T.Z. Baharav, A. Leshem, N. Bambos). *IEEE Transactions on Selected Areas in Information Theory*, 2(2):584-598, June 2021.

14. **Incidence of and Factors Associated with Prolonged and Persistent Postoperative Opioid Use in Children 0 to 18 Years of Age** (A. Ward, E. De Souza, D. Miller, E. Wang, E. Sun, N. Bambos, T. Anderson). *Journal of Anesthesia & Analgesia 131(4):1237-1248, Oct. 2020.*

15. **On the Convergence of Mirror Descent Beyond Stochastic Convex Programming** (with Z. Zhou, P. Mertikopoulos, S. Boyd, P. Glynn). *SIAM Journal of Optimization, 30(1):687-716, 2020.*

16. **Clustering Users by Their Mobility Behavioral Patterns** (with I. Ben-Gal, S. Weinstock, G. Singer). *ACM Transactions on Knowledge Discovery from Data*, 13(4)45:1-28, 2019.

17. **Deterministic and Stochastic Wireless Network Games: Equilibrium, Dynamics and Price of Anarchy** (with Z. Zhou, P. Glynn). *Operations Research*, 66(6):1498-1516, 2018.

18. **Infinite Time Horizon Maximum Causal Entropy Inverse Reinforcement Learning** (with Z. Zhou, M. Bloem). *IEEE Transactions on Automatic Control*, 63(9): 2787-2802, September 2018.

19. **Stochastic Models of Ground Delay Program Implementation for Prediction, Simulation and Insight** (with M. Bloem). *Journal of Aerospace Operations*, 5(1-2):85-117 (2017), April 2018.

20. **Video Streaming Schemes for Industrial IoT** (with H. Kanzaki, K. Schubert). *Journal of Reliable Intelligent Environments*, 3(4):233-241, Dec. 2017.

21. **Improving Predictions of Pediatric Surgical Durations with Supervised Learning** (with N. Master, Z. Zhou, D. Miller, D. Scheinker, P. Glynn). *Int. Journal of Data Science and Analytics*, 4:35-52, May 2017.

22. **Service Rate Control of Tandem Queues with Power Constraints** (with L. Xia, D. Miller, Z. Zhou). IEEE Transactions on Automatic Control, March 2017.

23. **Power Control for Packet Streaming with Head-of-Line Deadlines** (with N. Master). Journal of Performance Evaluation, 106:1-18, 2016.

24. **Power Optimization in Random Wireless Networks** (with A. Moustakas, P. Mertikopoulos). IEEE Transactions on Information Theory, 62(9):5030-5058, September 2016.

25. **Adaptive Prefetching in Wireless Computing** (with N. Master, A. Dua, D. Tsamis, J.P. Singh). IEEE Transactions on Wireless Communications, 15(5):3296-3310, 2016.

26. **Myopic Policies for Non-Preemptive Scheduling of Jobs with Decaying Value** (with N. Master). Journal of Probability in Engineering and Informational Sciences, November 2016.

27. **Maximum Weight Matching with Hysteresis in Overloaded Queues with Setups** (with C. Chan, M. Armony). Queueing Systems, 82:315-351, 2016,

28. **Cone Schedules for Processing Systems in Fluctuating Environments** (with K. Ross, G. Michailidis). IEEE Transactions of Automatic Control, 60(10):2710-2715, 2015.

29. **Personalized Monitors for Real-Time Detection of Physiological States** (with L. Chow, A. Gilman, A. Chander). International Journal of e-Health and Medical Communications, 5(4):1-19, October 2014.

30. **Air Traffic Control Area Configuration Advisories from Near-Optimal Distinct Paths** (with M. Bloem). Journal of Aerospace Information Systems*,* 11(11):764-784, November 2014.

31. **Ground Delay Program Analytics with Behavioral Cloning and Inverse Reinforcement Learning** (with M. Bloem). AIAA Journal of Aerospace Information Systems, 2015.

32. **Optimizing Intensive Care Unit Discharge Decisions with Patient Readmission***s* (with C. Chan, V.F. Farias, G. Escobar). Operations Research (INFORMS), 60(6):1323-1341, Jan. 2013.

33. **BEST-AP: Non-Intrusive Estimation of Available Bandwidth and its Application for Dynamic Access Point Selection** (with P. Dely, L. Chow, N. Bayer, H. Einsiedler, C. Peylo). Computer Communications (Elsevier), 39:78-91, Jan 2014.

34. **A Software-Defined Networking Approach for Handover Management with Real-Time Video in WLANs** (with P. Dely, A. Kassler, L. Chow, N. Bayer, H. Einsiedler, C. Peylo, D. Mellado, M. Sanchez). Journal of Modern Transportation (Springer), 21:58-65, March 2013.

35. **Network Assisted Mobile Computing with Optimal Uplink Query Processing** (with C. Chan, J. Singh). IEEE Transactions on Mobile Computing, 60(6):1323-1341, Jan. 2013.

36. **Optimizing Intensive Care Unit Discharge Decisions with Patient Readmissions** (with C. Chan, V.F. Farias, G. Escobar). Operations Research (INFORMS), 60(6):1323-1341, Jan. 2013.

37. **Power and Delay Aware Management of Packet Switches** (with L. Mastroleon, D. O'Neill, B. Yolken) IEEE Transactions on Computers, , 61(12):1789-1799, Dec. 2012.

38. **Optimal State Surveillance under Observation Budget Constraints** (with P. Bommannavar). Int. Journal on Advances in Intelligent Systems, 4(4):57-67, Dec. 2011.

39. **A Scalable Delay-Power Control Algorithm for Bandwidth Sharing in Wireless Networks** (with F. Baccelli and N. Gast). IEEE/ACM Transactions on Networking, 19(5):1458-1471, October 2011.

40. **An Integrated Approach to Security Risk Management for IT-Intensive Organizations** (with Jeff Mounzer and Tansu Alpcan). Journal of Information Assurance and Security, 6(2):115-123, 2011.

41. **Channel, Deadline, and Distortion Aware Scheduling of Video Streams over Wireless Links** (with A. Dua, C. Chan, J. Apostolopoulos). IEEE Trans. on Wireless Communications, 9(3):1001-1011, March 2010.

42. **A Characterization of Max-Min SIR-Balanced Power Allocation with Applications** (with S. Stanczak, M. Kaliszan). ACM Wireless Networks, 16: 2335-2347, Nov. 2010.

43. **Game Based Capacity Allocation for Utility Computing Environments** *(with* B. Yolken). Telecommunications Systems Journal, pp. 1-17, May 2010.

44. **Scheduling Algorithms for Broadcasting Media with Multiple Distortion Measures**, (with C. W. Chan, S. Wee, J. Apostolopoulos), IEEE Trans. on Wireless Communications, 8(8) 4188-4199, 2009.

45. **Adaptive Data-Aware, Utility-Based Scheduling in Resource-Constrained Systems** (with D. Vengerov, L. Mastroleon, D. Murphy). Journal of Parallel and Distributed Computing, 70:871-879, 2009.

46. **Projective Cone Scheduling (PCS) Algorithms for Maximal Throughput Packet Switches** (with K. Ross). IEEE/ACM Transactions on Networking, 17(3):976-989, June 2009.

47. **Joint Task Migration and Power Management in Wireless Computing** (with S. Gitzenis). IEEE Transactions of Mobile Computing. 8(9): 1189-1204, Sept. 2009.

48. **Content-Aware Playout and Packet Scheduling for Video Streaming over Wireless** (with Y. Li, A. Markopoulou, J. Apostolopoulos) IEEE Trans. on Multimedia, 10(5):885-895, Aug. 2008.

49. **Joint Transmitter Power Control and Mobile Cache Management in Wireless Computing** (with S. Gitzenis). IEEE Transactions of Mobile Computing, 7(4):498-512, April 2008.

50. **Transmission Power and Duration-Aware Playout Control for Packetized Media Streaming over Wireless Links** (with Y. Li). Wireless Communications and Mobile Computing, 8:309-326, Jan. 2008

51. **Downlink Wireless Packet Scheduling with Deadlines** (with A. Dua). IEEE Trans. on Mobile Computing, 6(12), pp. 1410-1425, 2007.

52. **TCP Performance Dynamics and Link-Layer Adaptation Based Optimization Methods for Wireless Networks** (with J. P. Singh, Y. Li, A. Bahai, B. Xu, G. Zimmerman). IEEE Trans. Wireless Communications, 6(5), pp. 1864-1878, 2007.

53. **Transmission Power and Duration-Aware Playout Control for Packetized Media Streaming over Wireless Links** (with Y. Li). Wireless Communications and Mobile Computing, 2008.

54. **Joint Power-Playout Control for Media Streaming over Wireless Links** (with Y.Li, A. Markopoulou, J. Apostolopoulos), IEEE Transactions in Multimedia, 2006.

55. **Optimal Processor Allocation to Differentiated Job Flows** (with G. Michailidis and K. Wasserman), Performance Evaluation, v. 63(1), pp. 1-14, Jan. 2006.

56. **Queueing Networks of Random Link Topology: Stationary Dynamics of Maximum Throughput Schedules** (with G. Michailidis), Queueing Systems Theory and Applications, v. 50(1), pp. 5-52, May 2005.

57. **A Fuzzy Reinforcement Learning Approach to Power Control in Wireless Transmitters** (with D. Vengerov and H. Berenji). IEEE Trans. SMC (Part B), v. 35-4, pp. 768-778, Aug. 2005.

58. **Queuing and Scheduling in Random Environments** (with G. Michailidis). Advances in Applied Probability, vol. 36 (1), 293-317, 2004.

59. **Scheduling Optical Traffic Bursts in Time-Domain Wavelength Interleaved Optical Networks** (with K. Ross, K. Kumaran, I. Saniee, I. Widjaja). IEEE Journal of Selected Areas in Communications, vol. 21(9), pp. 1441-1451, 2003.

60. **Queueing Dynamics and Maximal Throughput Scheduling in Switched Processing Systems** (with M. Armony). Queueing Systems Theory and Applications, vol. 44(3), pp. 209-252, 2003.

61. **On Stability of Queueing Networks with Job Deadlines** (with A. Ward). Journal of Applied Probability, vol. 40(2), pp. 293-304, 2003.

62. **On Parallel Queueing with Random Server Connectivities and Routing Constraints** (with G. Michailidis). Probability in the Engineering and Informational Sciences, vol. 16, pp. 185-203, 2002.

63. **Optimal Control of Parallel Queues with Batch Service** (with C. Xia, G. Michailidis, P. Glynn). Probability in the Engineering and Informational Sciences, vol. 16, pp. 289-307, 2002.

64. **Power Controlled Multiple Access** (PCMA) Schemes in High Performance Wireless Packet Networks (with S. Kandukuri). IEEE Personal Communications, vol. 9(3), pp.58-64, 2002.

65. **Dynamic Online Task Scheduling on Parallel Processors** (with C. Xia, G. Michailidis). Performance Evaluation, vol. 46, pp. 219-233, 2001.

66. **Globally Constrained Power Control Across Multiple Channels in Wireless Packet Networks** (with S. Kandukuri). ACM Mobile Networks, vol. 6(5), pp. 427-434, 2001. (invited paper)

67. **Channel Access Algorithms with Active Link Protection for Wireless Communication Networks with Power Control** (with S. Chen, G. Pottie). IEEE/ACM Transactions on Networking, vol. 8(5), pp. 583-597, 2000.

68. **On Synchronization of Poisson Processes and Queueing Networks with Service and Synchronization Nodes** (with B. Prabhakar, T. Mountford). Advances in Applied Probability, vol. 32(3), pp. 824-843, 2000.

69. **Power-Induced Time Division on Asynchronous Channels** (with J. Rulnick). ACM Wireless Networks, vol. 5(2), pp. 71-80, 1999. (invited paper)

70. **Towards Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues and Design Aspects**. IEEE Personal Communications, vol. 5(3), pp. 50-59, 1998. (invited paper)

71. **Mobile Power Management for Wireless Communication Networks** (with J. Rulnick). ACM Wireless Networks, vol. 3(1), pp. 3-14, 1997. (invited paper)

72. **On a Singular Feature of Critical G/M/1 Queues** (with B. Prabhakar). Systems and Control Letters, vol. 28(5), pp. 239-245, 1996.

73. **Wireless, Mobile, Multimedia Networks** (with A. Alwan, R, Bagrodia, N. Bambos, J. Cong, M. Gerla, L. Kleinrock, J. Villasenior). IEEE Personal Communications, vol. 3(2), pp. 34-51, 1996.

74. **Convergence of Departures in Tandem Networks of */GI/Infinity Queues** (with B. Prabhakar, T. Mountford). Probability in the Engineering and Informational Sciences, vol. 10(4), pp. 487-500, 1996.

75. **The Supercomputer Supernet Testbed: A WDM-Based Supercomputer Interconnect** (with L. Kleinrock, M. Gerla, J. Cong, E. Gafni, L. Bergman, J. Bannister, S.P. Monacos, T. Bujewski, P-C. Hu, B. Kannan, B. Kwan, E. Leonardi, J. Peck, P. Palnati, S. Walton). IEEE/OSA Journal of Lightwave Technology, vol. 14(6), pp. 1388-1399, 1996.

76. **Optimal Server Allocation to Parallel Queues with Finite Capacity Buffers** (with K. Wasserman). Probability in the Engineering and Informational Sciences, vol. 10(2), pp. 279-285, 1996.

77. **Asymptotic Optimality of Statistical Multiplexing in Pipelined Processing** (with K. Wasserman). Queueing Systems Theory and Applications, vol. 21, pp. 97-123, 1995.

78. **OPTIMIC: A Scalable, Distributed, All-Optical Terabit Network** (with L. Kleinrock, J. Cong, E. Gafni, M. Gerla, L. Bergman, J. Bannister). Journal of High Speed Networks, vol. 4(4), pp. 407-424, 1995.

79. **On Infinite Queueing Tandems** (with B. Prabhakar). Systems and Control Letters, vol. 23, pp. 305-314, 1994.

80. **Optimality Aspects of Greedy Schemes in Parallel Processing of Random Graph-Structured Jobs** (with S. Chen). Probability in the Engineering and Informational Sciences, vol. 8, pp. 229-243, 1994.

81. **On Stationary Tandem Queueing Networks with Job Feedback** (with K. Wasserman). Queueing Systems Theory and Applications, vol. 15, pp. 137-164, 1994.

82. **Scheduling and Stability Aspects of a General Class of Parallel Processing Systems** (with J. Walrand). Advances in Applied Probability, vol. 25, pp. 176-202, 1993.

83. **On Closed Ring Queueing Networks**. Journal of Applied Probability, vol. 29(4), pp. 979-995, 1992.

84. **On Flows in Stochastic Marked Graphs** (with F. Baccelli & J. Walrand). Probability in the Engineering and Informational Sciences, vol. 5, pp. 145-157, 1991.

85. **On Stability and Performance of Parallel Processing Systems** (with J. Walrand). Journal of the Association for Computing Machinery, vol. 38(2), pp. 429-452, 1991.

86. **An Invariant Distribution for the G/G/1 Queueing Operator** (with J. Walrand). Advances in Applied Probability, vol. 22, pp. 254-256, 1990.

87. **On the Asymptotic Execution Time of Multi-Tasked Processes on Tandem Processors** (with J. Walrand). Systems and Control Letters, vol. 13, pp. 391-396, 1989.

88. **On State-Dependent Queues and Acyclic Queueing Networks** (with J. Walrand). Advances in Applied Probability, vol. 21, pp. 681-701, 1989.

89. **On Queues with Periodic Inputs** (with J. Walrand). Journal of Applied Probability, vol. 26, pp. 381-389, 1989.

PEER-REVIEWED CONFERENCE PAPERS:

1. **Power-Optimized Processor slowdown Control** (A.J. Mann, N. Bambos). *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2022.

2. **Learning in Games with Quantized Payoff Observations** (K. Lotidis, P. Mertikopoulos, N. Bambos). *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2022.

3. **Framed Projective Cone Scheduling: Latency vs. Context-Switching Tradeoff in Data Centers** (E. Zeger, A.J. Mann, N. Bambos). *Proceedings of the IEEE Global Communications Conference*, Rio de Janeiro, Brazil, Dec. 2022.

4. **Queue Up your Regrets: Achieving the Dynamic Capacity Region of Multi-player Bandits** (I. Bistritz, N. Bambos). *Proceedings of the Conference on Neural Information Processing Systems*

(NeurIPS), 2022 (accepted).

5. **Active Testing for an Emerging Epidemic** (A.J. Mann, I. Bistritz, N. Bambos). *Proceedings of the IEEE International Conference on e-Health Networking, Applications & Services (HealthCom)*, 2022 (accepted).

6. **Online Learning for Load Balancing of Unknown Monotone Resource Allocation Games** (I. Bistritz, N. Bambos). *Proceedings of the 38th International Conference on Machine Learning (ICML)*, PMLR 139:968-979, July 2021.

7. **Controlling Epidemics via Testing** (K. Lotidis, A. Moustakas, N. Bambos). *Proceedings of the 2021 60th Conference on Decision and Control (CDC)*, pp. 2092-2097, Austin, TX, Dec. 2021.

8. **An OpenAI-OpenDSS Framework for Reinforcement Learning on Distribution-Level Microgrids** (K. Moy, C. Tae, Y Wang, G. Henri, N. Bambos, R. Rajagopal). *Proceedings of the 2021 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1-5, Washington, DC, July 2021.

9. **Power Controlled Slowdown in Data Centers** (A. Mann, N. Bambos). *Proceedings of IEEE International Conference on Communications (ICC)*, pp. 1-6, Korea, May 2022.

10. **Cooperative Multi-Player Bandit Optimization** (I. Bistritz, N. Bambos). Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), 22 pages, 2020.

11. **Distributed Distillation for On-Device Learning** (I. Bistritz, A. Mann, N. Bambos). Proceedings of the Conference on Neural Information Processing Systems (NeurIPS), 26 pages, 2020.

12. **My Fair Bandit: Distributed Learning of Max-Min Fairness with Multi-Player Bandits** (I. Bistritz, T.Z. Baharav, A. Leshem, N. Bambos). Proceeding of the International Conference on Machine Learning (ICML), PMLR 119: 930-940, 10 pages, 2020.

13. **Distributed Scheduling of Charging for On-Demand Electric Vehicle Fleets** (I. Bistritz, M. Klein, N. Bambos, O. Maimon, R. Rajagopal). Proceedings of the IFAC Workshop on Discrete Event Systems (WODES), 6 pages, 2020. [**Best student paper finalist award**; 1 best paper award and 3 finalist awards out of 75 papers presented at the conference.]

14. **Operationally-Informed Hospital-Wide Discharge Prediction Using Machine Learning** (by A. Ward, A. Mann, J. Vallon, G. Escobar, N. Bambos, A. Schuler). *Proceedings of the 22nd IEEE Int. Conf. on e-Health Applications & Services (HealthCom), 6 pages, Shenzhen, China, Dec. 2020.* [**Best Paper Runner-Up Award**; 1 best paper award and 2 runner-up awards out of 100 papers presented at the conference.]

15. **Rates of, and Risk Factors for, *Substance Use Disorder* in Children and Young Adults Before and After Surgery** (I. Thapa, A. Ward, B. De Souza, N. Bambos, T. Anderson). *Proceedings of the International Anesthesia Research Society 2020 Annual Meeting and International Science Symposium, San Francisco, CA, 2020; extended abstract published in the Journal of Anesthesia & Analgesia Supplement, 130(5)S:689-690.*

16. **Rates of, and Risk Factors for, Chronic Pain in Children and Young Adults Before and After Surgery (**I. Thapa, A. Ward, B. De Souza, N. Bambos, T. Anderson). *Proceedings of the International Anesthesia Research Society 2020 Annual Meeting and International Science Symposium, San Francisco, CA, 2020; extended abstract published in the Journal of Anesthesia & Analgesia Supplement, 130(5)S:691.*

17. **Development and Validation of a Machine Learning Algorithm for Prediction of Prolonged Opioid Use After Surgery in Adolescents and Young Adults** (Trisha Jani, A. Ward, B. De Souza, N. Bambos, Thomas A Anderson). *Proceedings of the International Anesthesia Research Society 2020 Annual Meeting and International Science Symposium, San Francisco, CA, 2020; extended abstract published in the Journal of Anesthesia & Analgesia Supplement, 130(5)S:712-713.*

18. **Distributed Scheduling of Charging for On-Demand Electric Vehicle Fleets** (with I. Bistritz, M. Klein, O. Maimon, R. Rajagopal). Proceedings of the 2020 Workshop on Discrete-Event Systems (WODES), Rio de Janeiro, Brazil, Nov. 2020.

19. **Online EXP3 Learning in Adversarial Bandits** (with I. Bistritz, Z. Zhou, X. Chen, J. Blanchet). Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS), Vancouver, Canada, December 2019.

20. **Controlling Contact Network Topology to Prevent Measles Outbreaks** (with I. Bistritz, D. Kahana, I. Ben-Gal, D. Yamin). Proceedings of the 2019 IEEE Global Communications Conference (Globecom), pp. 1-6, Waikoloa, Hawaii, December 2019.

21. **Asymptotically Optimal Distributed Gateway Load-Balancing for the Internet of Things** (with I. Bistritz). Proceedings of the 2019 IEEE International Conference on the Network of the Future (NoF), pp. 98-101, Rome, Italy, October 2019.

22. **Noninvasive Identification of Hypotension Using Convolutional-Deconvolutional Networks** (with D. Miller, A. Ward, A. Shin, D. Scheinker). Proceedings of 2019 IEEE International Conference on E-Health Networking, Applications and Services (HealthCom), pp. 1-6, Bogota, Colombia, October 2019.

23. **Learning to Emulate an Expert Projective Cone Scheduler** (with A. Ward, N. Master). Proceedings of the 2019 IEEE American Control Conference (ACC), pp. 292-297, Philadelphia, PA, July 2019.

24. **Incidence of Persistent Opioid Use in Children after Surgery** (with T. Anderson, A. Ward, B. De Souza, E. Wang, D. Miller). Proceedings of the International Anesthesia Research Society 2019 Annual Meeting and International Science Symposium, May 17-20, Montreal, Quebec, Canada; published in the Journal: Anesthesia & Analgesia Supplement 128(5)S-354:764-765.

25. **Smart Greedy Distributed Allocation in Microgrids** (with I. Bistritz, A. Ward, Z. Zhou). Proceedings of the 2019 IEEE International Conference on Communications (ICC), pp. 1-6, Shanghai, China, May 2019.

26. **Anesthesiologist Surgery Assignments Using Policy Learning** (with A. Ward, Z. Zhou, Ellen Wang, D. Scheinker). Proceedings of the 2019 IEEE International Conference on Communications (ICC),6 pages, Shanghai, China, May 2019.

27. **Learning Health State Transition Probabilities via Wireless Body Area Networks** (with Tal Geller, Yair Bar David, Evgeni Khmelnitsky, Irad Ben-Gal, A. Ward, D. Miller). Proceedings of the 2019 IEEE International Conference on Communications (ICC), pp. 1-6, Shanghai, China, May 2019.

28. **The Power of Consensus: Optimal Distributed Multichannel Wireless Transmitter Power Control (**with I. Bistritz**).** Proceedings of the 2019 IEEE International Conference on Communications (ICC), pp. 1- 6, Shanghai, China, May 2019.

29. **Bidding-Based Dynamic Power Pricing Scheme in Smart Grids** (with A. Ward, Z. Zhou). Proceedings of the 2019 IEEE International Conference on Computing, Networking and Communications (ICNC), pp. 729-734, Honolulu, Hawaii, February 2019.

30. **Learning in Games with Lossy Feedback** (with Z. Zhou, P. Mertikopoulos, S. Athey, P. Glynn and Y. Ye). Proceedings of 32nd Neural Information Processing Systems (NIPS) Conference, 11 pages, Montreal, Canada, December 2018.

31. **Distributed Asynchronous Stochastic Optimization under Unbounded Delays: How Slow Can You Go?** (with Z. Zhou, P. Mertikopoulos, P. Glynn and Y. Ye, Li-Jia Li, Li Fei-Fei). Proceedings of the 35th International Conference on Machine Learning (ICML), 10 pages, Stockholm, Sweden, July 2018.

32. **Power Control with Random Delays: Robust Feedback Averaging** (with A. Ward, Z. Zhou, P. Mertikopoulos). Proceedings of the 57th Conference on Decision and Control (CDC), pp. 7040-7045, Miami Beach, FL, December 2018.

33. **Robustness of Join-the-Shortest-Queue Scheduling to Communication Delay** (with S. Mehdian, Z. Zhou). Proceedings of the 2018 American Control Conference (ACC), pp. 3708-3713, Milwaukee, WI, June 2018.

34. **Optimal Health Monitoring via Wireless Body Area Networks** (with Yair Bar David, Tal Geller, Evgeni Khmelnitsky, Irad Ben-Gal, A. Ward, D. Miller). Proceedings of the 57th Conference on Decision and Control (CDC), pp. 7040-7045, Miami Beach, FL, December 2018.

35. **Sensing-Constrained Power Control in Digital Health** (with D. Miller, Z. Zhou, N. Bambos, I. Ben-Gal). Proceedings of the 2018 American Control Conference (ACC), pp. 3708-3713, Milwaukee, WI, June 2018.

36. **Optimal Sensing for Patient Health Monitoring** (with D. Miller, Z. Zhou, I. Ben-Gal). Proceedings of the 2018 International Communications Conference (ICC), 7 pages, Kansas City, MO, May 2018.

37. **Physiological Waveform Imputation of Missing Data using Convolutional Autoencoders** (with D. Miller, A. Ward, D. Scheinker, Andrew Shin). Proceedings of the 2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom), pages 6, Ostrava, Czech Republic, September 2018.

38. **Automatic Sleep Arousal Identification from Physiological Waveforms Using Deep Learning** (with D. Miller, A. Ward, N. Bambos). Proceedings of 2018 Computing in Cardiology (CinC) Conference, pages 5, Maastricht, Netherlands, September 2018.

39. **Bidding-Based Dynamic Power Pricing Scheme in Smart Grids** (with A. Ward, Z. Zhou). Proceedings of the 2018 IEEE International Conference on Computing, Networking and Communications (ICNC), pp. 729-734, Honolulu, Hawaii, February 2019.

40. **Smart Greedy Distributed Allocation in Microgrids** (with I. Bistritz, A. Ward, Z. Zhou). Proceedings of the 2019 IEEE International Conference on Communications (ICC), pages 6, Shanghai, China, May 2019 (accepted 2018).

41. **Anesthesiologist Surgery Assignments Using Policy Learning** (with A. Ward, Z. Zhou, Ellen Wang, D. Scheinker). Proceedings of the 2019 IEEE International Conference on Communications (ICC), pages 6, Shanghai, China, May 2019 (accepted 2018).

42. **Learning Health State Transition Probabilities via Wireless Body Area Networks** (with Tal Geller, Yair Bar David, Evgeni Khmelnitsky, Irad Ben-Gal, A. Ward, D. Miller). Proceedings of the 2019

IEEE International Conference on Communications (ICC), pages 6, Shanghai, China, May 2019 (accepted 2018).

43. **The Power of Consensus: Optimal Distributed Multichannel Wireless Transmitter Power Control** (with I. Bistritz**).** Proceedings of the 2019 IEEE International Conference on Communications (ICC), pages 6, Shanghai, China, May 2019 (accepted 2018).

44. **Countering Feedback Delays in Multi-Agent Learning** (Z. Zhou, P. Mertikopoulos, N. Bambos, P. Glynn, C. Tomlin), Proceedings of Advances for Neural Information Processing Systems (NIPS), pp.1-11, Long Beach, CA, Dec. 2017.

45. **Stochastic Mirror Descent in Variationally Coherent Optimization Problems** (with Z. Zhou, P. Mertikopoulos, S. Boyd, P. Glynn), Proceedings of Advances for Neural Information Processing Systems (NIPS), pp. 1-11, Long Beach, CA, Dec. 2017.

46. **Classical Communication Network Design Via Quantum Heuristics** (with A. Ward, N. Master). Proceeding of the Conference on Decision and Control (CDC), pp. 1-6, Long Beach, CA, Dec. 2017.

47. **Mirror Descent Learning in Continuous Games** (with Z. Zhou, P. Mertikopoulos, A. Moustakas, P. Glynn). Proceeding of the Conference on Decision and Control (CDC), pp. 1-6, Long Beach, CA, Dec. 2017.

48. **Stable Power Control in Wireless Networks via Dual Averaging** (with Z. Zhou, P. Mertikopoulos, A. Moustakas, S. Mehdian, Peter Glynn). Proceedings of IEEE Global Communications Conference (GLOBECOM), pp. 1-6, Singapore, Dec. 2017.

49. **Longest-Queue-First Scheduling with Intermittent Sampling** (with S. Mehdian, Z. Zhou). Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6, Montreal, QC, Canada, Oct. 2017.

50. **Least Action Routing: Identifying the Optimal Path in a Wireless Relay Network** (with A. Moustakas, P. Mertikopoulos, Z. Zhou). Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6, Montreal, QC, Canada, Oct. 2017.

51. **Market-Based Dynamic Service Mode Switching in Virtualized Wireless Networks** (with M. Dimakopoulou, M. Valdez-Vivas, J. Apostolopoulos). Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1-6, Montreal, QC, Canada, Oct. 2017.

52. **Video Streaming Services for Industrial IoT** (with H. Kanzaki, K. Schubert). Proceedings of International Conference on Computer Communications and Networks (ICCCN), pp. 1-7, Vancouver, CA, Aug. 2017.

53. **Dynamic Control of Data Center Network and Computation Resources** (with D. Miller, Li Xia, Z. Zhou, N. Bambos). Proceedings of the International Conference on Computing, Networking and Communications (ICNC), pp. 1-6, San Jose, CA 2017.

54. **Asynchronous Best Response Dynamics for Resource Allocation Games in Utility Computing** (with K. Schubert, N. Master, Z. Zhou). Proceedings of IEEE American Control Conference (ACC), 6 pages, Seattle, WA, May 2017.

55. **An Infinite Dimensional Model for a Single-Server Priority Queue** (with N. Master, Z. Zhou). Proceedings of IEEE American Control Conference (ACC), 6 pages, Seattle, WA, May 2017.

56. **Join-the-Shortest Queue Scheduling with Delay: A Case Study** (with S. Mehdian, Z. Zhou). Proceedings of IEEE American Control Conference (ACC), 6 pages, Seattle, WA, May 2017.

57. **Cost-Sensitive Security Risk Management for Large-Scale Computing Infrastructures** (with N. Master). Proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC), 6 pages, San Jose, CA, January 2017.

58. **An Infinite Dimensional Model for a Many Server Priority Queue** (with N. Master, Z. Zhou). Proceeding of the Conference on Information Sciences and Systems (CISS), 6 pages, Johns Hopkins University, Baltimore, MD, March 2017.

59. **Wireless Channel Selection with Non-Invasive Power Probing** (with K. Schubert). Proceedings of IEEE Wireless Telecommunications Symposium, 6 pages, Chicago, IL, April 2017.

60. **Infinite Server Queueing Networks with Deadline Bases Routing** (with N. Master), IEEE Conference on Decision and Control (CDC), pp. 5360-5366, Las Vegas, NV, December 2016.

61. **Repeated Games for Power Control in Wireless Communications: Equilibrium and Regret** (with Z. Zhou, P. Glynn). Proceedings of IEEE Conference on Decision and Control (CDC), pp. 3603-3610, Las Vegas, NV, December 2016.

62. **A Game-Theoretical Formulation of Influence Networks** (w Z. Zhou, B. Yolken, R.A. Miura-Ko). Proceeding of IEEE American Control Conference, pp. 3802-3807, Boston, MA, July 2016.

63. **Dynamics of Linear Influence Networks under Stochastic Environments** (with Z. Zhou, P. Glynn). Proceeding (Springer) of Conference on Decision and Game Theory for Security (GameSec), 6 pages, New York, NY, November 2016.

64. **A Stochastic Stability Characterization of the Foschini-Miljanic Algorithm in Random Wireless Networks** (with Z. Zhou, D. Miller, P. Glynn). Proceedings of IEEE Global Communications Conference (GLOBECOM), 6 pages, Washington, DC, December 2016.

65. **Wireless Power-Controlled TCP with Holdover** (with K. Schubert). Proceedings of IEEE International Conference on Communications (ICC), 6 pages, Kuala Lumpur, Malaysia, May 2016.

66. **Data Aggregation for Low Power Wireless Devices** (with K. Schubert). Proceedings of IEEE MILCOM, 6 pages, November 2016.

67. **Detecting Inaccurate Predictions of Pediatric Surgical Durations** (with Z. Zhou, D. Miller, N. Master, P. Glynn). IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 452-457, Montreal, Canada, October 2016.

68. **Reliable and Efficient Performance Monitoring in Linux** (with M. Dimakopoulou, S. Eranian, N. Koziris). Proceedings of International Conference for High Performance Computing, Networking, Storage and Analysis, 11 pages, Salt Lake City, Utah, November 2016.

69. **Scalable Data Center Power Management via a Global Stress Signal** (with D. Miller, N. Master, Z. Zhou, N. Bambos). IEEE Global Communications Conference (GLOBECOM), 7 pages, 2015.

70. **Target-Rate Driven Resource Sharing in Queueing Systems** (with Z. Zhou). Proceedings of IEEE Conference on Decision and Control (CDC), pp. 4490-4945, Osaka, Japan, December 2015.

71. **Wireless Communication Games in Fixed and Random Environments** (with Z. Zhou). Proceedings of IEEE Conference on Decision and Control (CDC), pp. 1637-1642, Osaka, Japan, December 2015.

72. **Dynamic Management of Network Risk from Epidemic Phenomena** (A. Sinha, J. Ducci). IEEE Conference on Decision and Control (CDC), 6 pages, Osaka, Japan, December 2015.

73. **Low Latency Policy Iteration via Parallel Processing and Randomization** (with N. Master). IEEE Conference on Decision and Control (CDC), pp. 1084-1091, 2015.

74. **Channel Responsive Wireless TCP** (with K. Schubert, H. Endo). IEEE International Communications Conference (ICC'14), 6 pages, London, UK, 2015.

75. **A General Model for Resource Allocation in Utility Computing** (with Z. Zhou). American Control Conference, pp. 1746-1751, Chicago, IL, 2015.

76. **Service Rate Control for Jobs with Decaying Value** (with N. Master). American Control Conference, 6 pages, Chicago, IL, 2015.

77. **Ground Delay Program Analytics with Behavioral Cloning and Inverse Reinforcement Learning** (with M. Bloem). Proceedings of AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 6 pages, Atlanta, GA, June 2014.

78. **Infinite Time Horizon Maximum Causal Entropy Inverse Reinforcement Learning** (with M. Bloem). IEEE Conference on Decision and Control (CDC'14), pp. 4911-4916, Los Angeles, CA, December 2014.

79. **Power-Controlled Multiple Access with Queue-Dependent Backoff Threshold** (with J. Mounzer, K. Schubert, A. Goldsmith). IEEE Golbal Communications Conference (Globecom'14), pp. 4738-4744, Austin, TX, December 2014.

80. **Randomized Iterations for Low Latency Fixed Point Computation** (with N. Master). IEEE Conference on Decision and Control (CDC'14), pp. 5208-5215, Los Angeles, CA, December 2014.

81. **Distributed Smart Grid Architecture for Delay and Price Sensitive Power Management** (with N. Master, J. Mounzer), accepted to the IEEE International Communications Conference (ICC'14), pp. 3670-3675, Sydney, Australia, June 2014.

82. **Dynamic Resource Management in Virtualized Data Centers with Bursty Traffic** (with M. Valdez-Vivas, J. Apostolopoulos), accepted to the IEEE International Communications Conference (ICC'14), Sydney, Australia, June 2014.

83. **Power Control for Wireless Streaming with HOL Packet Deadlines** (with N. Master), accepted to the IEEE International Communications Conference (ICC'14), pp. 2263-2269, Sydney, Australia, June 2014.

84. **Channel Exploration for Wireless Media Streaming with Handoff and Rebuffering Control** (with L. Chow, J.P. Singh), accepted to the IEEE International Communications Conference (ICC'14), pp. 2281-2288, Sydney, Australia, June 2014.

85. **Channel Aware Rebuffering for Wireless Media Streaming with Handoff Control** (with L. Chow, B. Collins, P. Dely, A. J. Kassler, C. Peylo, H. J. Einsiedler, N. Bayer). Proceedings of the 2013 IEEE Global Communications Conference (GlobeCom'13), pp. 4434-4439, Atlanta, Georgia, December 2013.

86. **Congestion versus Accuracy Tradeoffs in IP Traffic Classification** (with M. Valdez-Vivas). Proceedings of the 2013 IEEE Global Communications Conference (GlobeCom'13), Atlanta, Georgia,

December 2013.

87. **Playout Buffer Responsive Wireless Streaming for Multiple Clients** (with P. Bommannavar, J. Apostolopoulos). Proceedings of the 2013 IEEE Global Communications Conference (GlobeCom'13), pp. 2276-2281, Atlanta, Georgia, December 2013.

88. **QoS-Aware Admission Control in Heterogeneous Data Centers** (with C. Delimitrou, C. Kozyrakis). Proceedings of USENIX International Conference on Autonomic Computing, San Jose, CA, June 2013.

89. **Delay-Sensitive Power Management for Packet Switches** (with M. Valdez-Vivas, D. O'Neill). Proceedings of IEEE International Communications Conference (ICC'13), Budapest, Hungary, June 2013.

90. **Latency and Energy in Quality-Driven Applications for Networked Wireless Devices** (with M. Valdez-Vivas). Proceedings of IEEE International Communications Conference (ICC'13), Budapest, Hungary, June 2013.

91. **Resource Allocation and Scheduling for Energy Efficient Tracking** (with P. Bommannavar, J. Apostolopoulos). Proceedings of IEEE International Communications Conference (ICC'13), pp. 1532-1537, Budapest, Hungary, June 2013.

92. **Deadline Aware Packet Scheduling in Switches for Multimedia Streaming Applications** (with P. Bommannavar, J. Apostolopoulos). Proceedings of IEEE International Communications Conference (ICC'13), pp. 3797-3802, Budapest, Hungary, June 2013.

93. **Power Optimization in Random Wireless Networks (with A.L. Moustakas).** Proceedings of IEEE International Symposium on Information Theory (ISIT'13), Istanbul, Turkey, July 2013.

94. **Real-Time Physiological Stream Processing for Health Monitoring Services** (with L. Chow). Proceedings of International Conference on e-Health Networking, Applications and Services (HealthCom'13), pp. 611-616, Lisbon, Portugal, Oct. 2013.

95. **An Approach for Finding Multiple Areas of Specialization Configuration Advisories** (with M. Bloem). Proceedings of AIAA Aviation Technology, Integration, and Operations (ATIO), 6 pages, Los Angeles, CA, August 2013.

96. **Playout-Buffer Aware Hand-Off Control for Wireless Video Streaming** (with L. Chow, B. Collins, C. Peylo, H. Einsiedler, N. Bayer, P. Dely, A. Kassler), IEEE Global Communications Conference (GlobeComm), pp. 5237-5242, Los Angeles, CA, 6 pages, Dec. 2012.

97. **Resource Constrained Failure Management in Networked Computing Systems** (with P. Bommannavar). IEEE Global Communications Conference (GlobeComm), pp. 1884-1889, Los Angeles, CA, 6 pages, Dec. 2012.

98. **Approximating the Likelihood of Historical Airline Actions to Evaluate Airline Delay Cost Functions** (with M. Bloem, H. Huang). IEEE Conference on Decision and Control (CDC), Maui, HI, pp. 508-513, Dec. 2012.

99. **Power Optimization on a Network: The Effect of Randomness** (with A. Moustakas), IEEE Int. Symposium of Information Theory, pp. 606-610, June 2012.

100. **Nonlinear Cooperative Dynamics in Distributed Power Control for Wireless Networks** (with J. Mounzer). IEEE Int. Communications Conference (ICC), pp. 5042-5047, June 2012.

101. **Evaluation of Algorithms for a Miles-in-Trail Decision Support Tool** (with M. Bloem, D. Hattaway, and N. Bambos). Int. Conference on Research in Air Transportation, Berkeley, CA, 8 pages, May 2012.

102. **Power Budgeted Packet Scheduling for Wireless Multimedia** (with P. Bommannavar, J. Apostolopoulos). International Communications Conference (ICC), pp. 2108-2113, June 2012.

103. **Power Control in Random Networks: The Effect of Disorder in User Positions** (with A. Moustakas). IEEE Int. Workshop for Spatial Stochastic Models for Wireless Networks (SPASWIN), pp. 380-385, May 2012.

104. **Intrusion Detection in IT Networks with Limited Observations** (with P. Bommannavar) IEEE International Conference on Computing, Networking and Communications (ICNC), Maui, Hawaii, pp. 6, Nov. 2012.

105. **Coordinated Tactical Air Traffic and Airspace Management** (with M. Bloem) IEEE Conference on Decision and Control (CDC), pp. 5287-5292, Orlando, FL, Dec. 2011.

106. **Implementation of Fine-Grained Power/Bitrate Control for 802.11 Devices** (with Min-Wook Jeon, A. Kabani). Int. Tech. Conference on Circuits/Systems, Computers and Communications, Gyeonju, Korea, June 2011.

107. **Security Risk Management in Computing Systems with Constraints on Service Disruption** (with P. Bommannavar), IEEE International Conference on Computer Communication Networks (ICCCN), pp. 6, Maui, Hawaii, June 2011.

108. **Security Risk Management via Dynamic Games with Learning** (with Praveen Bommannavar and Tansu Alpcan. IEEE Int. Communications Conference (ICC), pp. 1-6, Kyoto, Japan, June 2011.

109. **Dynamic Control and Mitigation of Interdependent IT Security Risks** (with J. Mounzer, T. Alpcan) IEEE Int. Communications Conference (ICC), pp. 1-6, Cape Town, South Africa, May 2010 (Best Paper Award).

110. **Optimal State Surveillance under Budget Constraints** (with P. Bommannavar) Int. Conf. on Emerging Network Intelligence (Emerging`10), pp. 1-6, Florence, Italy, Oct. 2010 (Best Paper Award).

111. **Hybrid Power Control Algorithms for Streaming and Data Traffic in Wireless Networks** (with J. Mounzer) IEEE Int. Communications Conference (ICC), pp. 1-6, Cape Town, South Africa, May 2010.

112. **Integrated Security Risk Management for IT-Intensive Organizations** (with J. Mounzer, T. Alpcan) Sixth Int. Conf. on Information Assurance and Security (IAS), pp. 329-334, Atlanta, GA, Aug. 2010.

113. **Locking Dynamics and Mitigation Schemes in Distributed Power Control for Wireless Networks** (with J. Mounzer) IEEE Global Communications Conference (GlobeComm), pp. 1-6, Dec. 2010 .

114. **Fairness of Heterogeneous Queues through Cone Scheduling** (with C. Chan, M. Armony). Manufacturing and Service Operations Management Society Conference, pp. 1-3, Technion, Haifa, Israel, June 2010.

115. **Patch Scheduling for Risk Exposure Mitigation under Service Disruption Constraints** (with P. Bommannavar) 4th Int. Conf. on Emerging Security Information Systems and Technologies (SecureWare), pp. 176-181, Venice, July 2010.

116. **Admission Control for Autonomous Wireless Links with Power Constraints** (with M. Kaliszan, S. Stanczak). IEEE Int. Conf. in Acoustics Speech and Signal Processing, pp. 3138-3141, Dallas, TX, March 2010.

117. **Backlog aware scheduling for ingress memories in high-radix, single-stage switches** (with D. Tsamis, B. Yolken, W. Olesinski, H. Eberle, N. Gura). IEEE Global Com. Conference (Globecom), Honolulu, HI, pp. 1-6, Dec. 2009

118. **Throughput Loss in Task Scheduling due to Server State Uncertainty** (with C. Chan). ACM Int. Conf. in Performance Evaluation Methodologies and Tools (ValueTools), pp. 1-10, Pisa, Italy, 2009.

119. **Game Theoretic Rate Control for Mobile Devices** (with D. Tsamis, T. Alpcan). IEEE Int. Conf. on Game Theory for Networks (Gamenets), Instabul, Turkey, 2009.

120. **Dynamic Resource Modeling for Heterogeneous Wireless Networks** (with D. Tsamis, T. Alpcan, J. Singh). IEEE Int. Conf. on Communications (ICC), Dresden, Germany, pp. 1-6, 2009.

121. **Backlog aware scheduling for ingress memories in high-radix, single-stage switches** (with D. Tsamis, B. Yolken, W. Olesinski, H. Eberle, N. Gura). IEEE Global Com. Conference (Globecom), Honolulu, HI, pp. 1-6, 2009.

122. **Cost and Target-Based Scheduling for Switch Power Control** (with B. Yolken, D. Tsamis). IEEE International Conference on Communications (ICC), Dresden, Germany, 2009.

123. **Modeling Dependencies in Security Risk Management** (with T. Alpcan) ACM Int. Conf. on Risk and Security of Internet and Systems (CRiSIS), Toulouse, France, 6 pgs, 2009.

124. **A Characterization of Max-Min Power Allocation with Applications** (with Slawomir Stanczak, Michal Kaliszan, Marcin Wiczanowski). IEEE Int. Symposium on Information Theory, Seoul, Korea, 6 pgs, June 2009.

125. **Maximizing Throughput of Hospital Intensive Care Units with Patient Readmissions** (with C. W. Chan, V. F. Farias, G. J. Escobar). INFORMS Manufacturing and Service Operations Management (MSOM) Conference, Boston, MA, 5 pgs, June 2009.

126. **Target-Based Power Control for Queueing Systems with Applications to Packet Switches** (with B. Yolken, D. Tsamis). IEEE Global Telecom. Conference (Globecomm), New Orleans, pp. 1-6, Dec. 2008.

127. **Security Investment Games in Interdependent Organizations** (with A. Miura, A. Yolken, J. Mitchell), Proc. 46th Allerton Conference on Communication, Control and Computing, pp. 1-10, Sept. 2008.

128. **Game Based Admission Control for Wireless Systems** (with B. Yolken) Internationa Wireless Internet Conference (WICON), pp. 1-7, Nov. 2008.

129. **Admission Control for Power-Controlled Wireless Networks under General Interference Functions** (with S. Stanczak, M. Kaliszan), 42nd Asilomar Conference on Signals, Systems and Computers, *Monterey, USA, Oct. 2008*.

130. **Fast Power-Up Active Link Protection in Autonomous Distributed Transmitter Power Control** (with S. Gitzenis), Proc. IEEE International Wireless Communication and Computing Conference (IWCMC), pp. 1-6, Chania, Greece, Aug. 2008.

131. **Game Based Capacity Allocation for Utility Computing Environments** (with B. Yolken), Proc. GameComm, pg. 1-10, Athens, Greece, Oct. 2008.

132. **Network-Assisted Wireless Computing** (with C. Chan and J. P. Singh), Proc. IEEE PIMRC, 2008.

133. **Target-Based Power Control for Queueing Systems with Applications to Packet Switches** (with B. Yolken and D. Tsamis), IEEE Globecom, New Orleans, 2008.

134. **Security Decision-Making amongst Interdependent Organizations** (with A. Miura, B. Yolken, J. Mitchell), Proc. IEEE Computer Security Foundations Symposium (CSF), Pittsburgh, PA, 2008

135. **Wireless Broadcasting to Diverse Users** (with C.W. Chan, S. Wee, J. Apostolopoulos). Proc. IEEE International Conference on Communications, Beijing, China, 2008.

136. **Receiver Based Optimization of Video Delivery over Wireless Links** (with C.W. Chan, S. Wee, J. Apostolopoulos), Proc. IEEE International Conference on Multimedia, 2008.

137. **Backlog Aware Scheduling for Large Buffered Crossbar Switches** (A. Dua, B. Yolken, W. Olesinski, H. Eberle, N. Gura). Proc. IEEE International Communications Conference, Beijing, China, May 2008.

138. **Target-Driven and Incentive-Aligned Power Control for Wireless Networks** (with B. Yolken), Proc. IEEE Globecom, Washington, DC, Nov. 2007.

139. **Optimal Scheduling of Media Packets with Multiple Distortion Measures** (with C.W. Chan, S. Wee, J. Apostolopoulos), Proc. IEEE International Conference on Multimedia, Beijing, China, June 2007.

140. **Power Aware Management of Packet Switches** (with L. Mastroleon, D. ONeill, B. Yolken), Proc. IEEE Hot Interconnets, Stanford, CA, Aug. 2007.

141. **Dynamic Control Approach to Risk Mitigation in Computing Infrastructures** (with A. Miura-Ko). Proc. IEEE International Symposium on Information Assurance and Security, Manchester, UK, Aug. 2007.

142. **Backlog Aware Low Complexity Schedulers for Input Queued Packet Switches** (with A. Dua, W. Olesinksi, H. Eberle, N. Gura). Proc. IEEE Hot Interconncets, pp. 39-46, Stanford, CA, 2007.

143. **Performance Tradeoffs in Mobile Computing: To Fetch or Not to Fetch?** (with A. Dua, J. Singh) Proc. ACM MobiWac, Crete, Greece, Oct. 2007.

144. **Buffer Management in Wireless Media Streaming** (with A. Dua). Proc. IEEE Globecom, Washington, DC, Dec. 2007

145. **Distributed Backlog-Driven Power Control in Wireless Networking** (with A. Dua). Proc. IEEE LanMan, Princeton, NJ, June 2007.

146. **Power Management of Packet Switches via Differentiated Delay Targets** (with B. Yolken). Proc. IEEE International Communications Conference, Beijing, China, May 2007.

147. **Power-Managed Packet Switching** (with A. Dua, B. Yolken). Proceedings of IEEE International Communications Conference (ICC'07), Glasgow, UK, 2007.

148. **Wireless Packet Scheduling with Soft Deadlines** (with A. Dua). Proceedings of IEEE International Communications Conference (ICC'07), Glasgow, UK, 2007

149. **SecureRank: An Efficient Algorithm for Prioritizing Vulnerabilities in Computing Systems** (with R. A. Miura-Ko). Proceedings of IEEE International Communications Conference (ICC'07), Glasgow, UK, 2007.

150. **Job Scheduling for Maximal Throughput in Autonomic Computing Systems** (with K. Ross), Proc. of International Workshop of Self-Organizing Systems (IWSOS), Passau, Germany, September 2006 (and in Lecture Notes in Computer Science 4124 Springer 2006).

151. **Capacity Maximizing Packet Scheduling Algorithms for Interconnection Networks with Finite Buffers** (with K. Cross). IEEE Global Communications Conference (GlobeCom), San Francisco, CA, 2006.

152. **Receiver-Based Optimization for Video Delivery over Wireless Links** (with C. Chan, J. Apostolopoulos, Y. Li). Proceedings of IEEE International Conference on Multimedia (ICME'06), Toronto, Ontario, July 2006.

153. **Optimal Throughput and Routing for Wireless Link Arrays** (with F. Baccelli, C. Chan). Proceedings of Infocom 2006, 12 pages, Barcelona, Spain, April 2006.

154. **Joint Power Allocation and Scheduling for Deadline Constrained Wireless Traffic** (with A. Dua). Proceeding of IEEE Global Telecommunications Conference (GlobeCom'06), San Francisco, CA, San Francisco, Dec. 2006.

155. **Low-Jitter Scheduling Algorithms for Deadline-Aware Packet Switches** (with A. Dua) Low-Jitter Scheduling Algorithms for Deadline-Aware Packet Switches. Proceeding of IEEE Global Telecommunications Conference (GlobeCom'06), San Francisco, CA, Dec. 2006.

156. **Patching Rate Management for Controlled Service Disruption in Data Centers** (with L. Mastroleon, R. A. Miura-Ko). Proceeding of IEEE Global Telecommunications Conference (GlobeCom'06), San Francisco, CA, San Francisco, Dec. 2006.

157. **Packet Transmission and Content-Dependent Playout for Video Streaming over Wireless** (with Y. Li, A. Markopoulou, J. Apostolopoulos). Proceedings of IEEE Multimedia Signal Processing (MMSP'05), 10 pages, Shanghai, China, Oct. 2005.

158. **Power Management in Battery-Constrained Handheld Devices**, (with A. Markopoulou, Y. Li, C. Chan) Proceedings of IEEE International Conference on Broadband Networks (BroadNets'05), pp. 441-450, Boston, MA, Oct. 2005.

159. **Dynamic Risk Mitigation for 'Self-Defending' Network Security**, SecureComm, Athens, May 2005.

160. **On the Fairness-Delay Tradeoff in Wireless Packet Scheduling** (with A. Dua). Proceeding of IEEE Global Telecommunications Conference (GlobeCom'05), v. 5, pp. 2544-2548, St. Louis, MO, Nov. 2005.

161. **Power Control and QoS Tradeoffs for Real-Time Wireless Traffic** (with A. Dua). Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'06), vol. 2, pp. 1148-1153, Las

Vegas, NV, April 2006.

162. **Downlink Scheduling of Heterogeneous Traffic** (with A. Dua). Proceedings of IEEE International Conference on Communications (ICC'06), Instabul, Turkey, June 2006.

163. **Scheduling with Soft-Deadlines for Input-Queued Switches** (with A. Dua). Proc. of the 44th Allerton Conference on Communication, Computation and Control, Allerton, IL, September 2006.

164. **Cross-Layer Multi-hop Wireless Routing for Inter-vehicle Communication** (with J.P. Singh, B. Srinivasan, D. Clawin) Proceedings of IEEE International Conference on Network Testbeds and Research Infrastructures (TridentCom'06), Barcelona, Spain, March 2006.

165. **Autonomic Power Management Schemes for Internet Servers and Data Centers** (with L. Mastroleon, C. Kozyrakis, D. Economou) Proceedings of IEEE Global Communications Conference (GlobeCom'05), v. 2, pp. 943-947, St. Louis, MO, Nov. 2005.

166. **Power-Managed Block Level File Decryption in Wireless Network Computing** (with S.Gitzenis), Proceedings of IEEE International Symposium on Modeling and Optimization in Mobile, Ad-Hoc, and Wireless Networks (WiOpt'06), Boston, MA, April 2006.

167. **On the Singular Behavior of a Queuing System with Random Connectivity** (with G. Michailidis). Proceedings of the IEEE Conference on Decision and Control (CDC'05) , pp. 5348-5353, Seville, Spain, Dec. 2005

168. **A Control Formulation of the Network Security Problem via a Risk Management Approach**. Proceedings of the 2005 IEEE International Conference on Security and Privacy for Emerging Areas in Communication Networks, 3 pages, Athens, Greece, Sept. 2005.

169. **Packet Transmission and Content-Dependent Playout for Video Streaming over Wireless** (with Y. Li, A. Markopoulou, J. Apostolopoulos). Proceedings of IEEE Multimedia Signal Processing (MMSP'05), 10 pages, Shanghai, China, Oct. 2005.

170. **Power Management in Battery-Constrained Handheld Devices**, (with A. Markopoulou, Y. Li, C. Chan) Proceedings of IEEE International Conference on Broadband Networks (BroadNets'05), pp. 441-450, Boston, MA, Oct. 2005.

171. **Joint Power/Playout Control Schemes for Media Streaming over Wireless Links** (with Y. Li, A. Markopoulou, J. Apostolopoulos). Proceedings of 14th IEEE Packet Video Workshop, 12 pages, Irvine, CA, Dec. 2004.

172. **Optimizing Timeout-Based Sleep Algorithms** (with A. Markopoulou). IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN'05), UCLA, Los Angeles, CA, April 2005.

173. **Deadline Constrained Packet Scheduling in Wireless Networks** (with A. Dua). Proceedings of IEEE Vehicular Technology Conference VTC-F'05, v. 1, pp. 196-200, Dallas, TX, Sept. 2005.

174. **On the Fairness-Delay Tradeoff in Wireless Packet Scheduling** (with A. Dua). Proceeding of IEEE Global Telecommunications Conference (GlobeCom'05), v. 5, pp. 2544-2548, St. Louis, MO, Nov. 2005.

175. **Channel-State-Awareness Based Transmission Power Adaptation for Efficient TCP Dynamics in Wireless Networks** (with J.P. Singh, Y. Li) Proceedings of IEEE International Conference on

Communications (ICC'05), v. 5, pp. 3553-3559, Seoul, Korea, March 2005.

176. **Empirical Observations on Wireless LAN Performance in Vehicular Traffic Scenarios and Link Connectivity Based Enhancements for Multihop Routing** (with J.P. Singh, B. Shrinivasan, D. Clawin, Y. Yan). Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'05), v. 3, pp. 1676-1682, New Orleans, LA, March 2005

177. **Transmission Power and Playout Control for Packetized Audio Streaming over Wireless Links**, (with Y. Li) Proceedings of IEEE International Conference on Wireless Networks, Communications and Mobile Computing, Workshop on Mobility Management and Wireless Access (MobiWac'05), v. 1, pp. 55-62, Maui, Hawaii, June 2005.

178. **Autonomic Power Management Schemes for Internet Servers and Data Centers** (with L. Mastroleon, C. Kozyrakis, D. Economou) Proceedings of IEEE Global Communications Conference (GlobeCom'05), v. 2, pp. 943-947, St. Louis, MO, Nov. 2005.

179. **Media and Data Traffic Coexistence in Power-Controlled Wireless Networks** (with S. Gitzenis). Proceedings of ACM Wireless Multimedia Networking and Performance Modeling (WMuNeP'05), pp. 78-85, Montreal, Quebec, CA, Oct. 2005.

180. **Dynamic Quality of Service Control in Packet Switch Scheduling** (with K. Ross). Proceedings of IEEE International Conference on Communications (ICC'05), v. 1, pp. 396-401, Seoul, Korea, May 2005.

181. **Packet Scheduling Across Networks of Switches** (with K. Ross) Proceedings of IEEE International Conference on Networking (ICN'05), Reunion Island, April 2005.

182. **On the Singular Behavior of a Queuing System with Random Connectivity** (with G. Michailidis). Proceedings of the IEEE Conference on Decision and Control (CDC'05) , pp. 5348-5353, Seville, Spain, Dec. 2005

183. **Efficient Data Prefetching for Power-Controlled Wireless Packet Networks** (with S. Gitzenis). Proceedings of ACM/IEEE Mobile and Ubiquitous Networking Conference (MobiQuitous), pp.64-73, Boston, MA, August 22-25, 2004.

184. **Power-Controlled Packet Relays in Wireless Data Networks** (with S. Gitzenis). Proceedings of IEEE Global Communications Conference (GlobeCom), vol. 1, pp. 464-469, San Francisco, CA, December 1-5, 2003.

185. **Integrated Power Control for Circuit-Switched & Packet-Switched Traffic in Wireless Networks** (with S. Gitzenis). Proceedings of the IEEE Vehicular Technology Conference (VTC), vol. 4, pp. 2604-2609, Orlando, FL, Fall 2003.

186. **Power Control for TCP Adaptation to High-mobility Broadband Systems** (with J.P. Singh, K. Radermacher, V. Scharf-Katz). Proceedings of IEEE Vehicular Technology Conference (VTC), Los Angeles, CA, 9 pages, Fall 2004.

187. **Power-Controlled Wireless Links for Media Streaming Applications** (with Y. Li). Proceedings of the IEEE Wireless Telecommunications Symposium 2004, pp. 102-111, Pomona, CA, May 14-15, 2004

188. **Power-Controlled Media Streaming in Interference-Limited Wireless Networks** (with Y. Li). Proceedings of the 2004 IEEE International Conference in Broadband Networks, pp. 560-568, San

Jose, CA, October 25-29, 2004.

189. **Power-Controlled Wireless Links for Video Streaming Applications** (with Y. Li). Proceedings of the IEEE Vehicular Technology Conference (VTC), Los Angeles, CA, 9 pages, Sept. 26-29 2004.

190. **Local Search Scheduling Algorithms for Maximal Throughput in Packet Switches** (with K. Ross). Proceedings of the IEEE INFOCOM 2004, Hong Kong, China, 12 pages, March 7-11, 2004.

191. **Optimizing Quality of Service in Prioritized Packet Switch Scheduling** (with K. Ross). Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 5 pages, June 20-24, 2004.

192. **A Service Risk Management Approach to Capacity Protection in Optical Networks** (N. Bambos, S. Gitzenis, A. Miura, O. Gerstel, L. Paraschis). Proceedings of IEEE Workshop in Local and Metropolitan Area Networks (LANMAN'04), Mill Valley, CA, pp. 69-74, April 2004.

193. **Mobile-to-Base Task Migration in Wireless Computing** (S. Gitzenis, N. Bambos). Proceedings of IEEE Pervasive Communications Conference *PerCom* 2004, pp. 187-196, Orlando, FL, March 14-17, 2004.

194. **Distributed Power and Admission Control for Wireless Networks** (with T. Holliday, A. J. Goldsmith, P. Glynn), ISIT 2004.

195. **Dynamic Scheduling of Optical Data Bursts in Time-Domain Wavelength Interleaved Networks** (Kevin Ross, K. Kumaran, I. Saniee, I. Widjaja). Proceedings of the 11th IEEE Symposium on High-Performance Interconnects, Stanford, CA, pp. 108-113, 2003.

196. **Local/Remote Task Execution in Network Computing** (with Savvas Gitzenis). Proceedings of the International Conference on Internet Computing, IC '03, Las Vegas, NV, pp. 842-848, June 2003.

197. **Integrated Power Control for Circuit-Switched and Packet-Switched Traffic in Wireless Networks** (with S. Gitzenis). Proceedings of the IEEE Fall Vehicular Technology Conference VTC'03, Orlando, FL, vol. 4, pp. 2604-2609, Oct. 2003.

198. **Proposal and Demonstration of Link-Connectivity-Assessment-Based Enhancements to Routing in Mobile Ad-Hoc Networks** (with J. Singh, D. Clawin, B. Shrinivasan). Proceedings of the IEEE Fall Vehicular Technology Conference VTC'03, Orlando, FL, Oct. 2003.

199. **Power Management of Packet Switch Architectures with Speed Modes** (with D. O'Neill). Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, Monticello, IL, Oct. 2003. (invited paper)

200. **Distributed Power Control for Time Varying Wireless Networks: Optimality and Convergence** (with T. Holliday, A. J. Goldsmith, P. Glynn), Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, Monticello, IL, Oct. 2003. (invited paper)

201. **Capacity Protection/Restoration in Tree Optical Networks for Enterprise Content Distribution** (with S. Gitzenis, O. Gertsel, L. Paraschis). Proceedings of the Workshop "Protection and Restoration: from SONET/SDH to Next-Generation Networks" of IEEE Global Communications Conference (GlobeCom '03), San Francisco, CA, 2003.

202. **Power-Controlled Packet Relays in Wireless Data Networks** (with Savvas Gitzenis). Proceedings of the IEEE Global Communications International Conference (GlobeCom '03), San Francisco, CA, vol.

1, pp. 464-469, Dec. 2003.

203. **Learning Policies for Power Control in Wireless Networks**. (with D. Vengerov, H. Berenji). Proceedings of the IEEE Vehicular Technology Conference, Vancouver, BC, Canada, pp. 2390-2394, Oct. 2002.

204. **Power Efficient MAC Scheme Using Channel Probing in Multirate Wireless Ad-Hoc Networks** (with JW. Kim). Proceedings of IEEE Vehicular Technology Conference, Vancouver, BC, Canada, pp. 2382-2386, Oct. 2002.

205. **Geometry of Scheduling Algorithms for Packet Switches** (with K. Ross). Proceedings of the Allerton Conference on Communication, Control and Computing, University of University at Urbana-Chanpain, Monticello, Illinois, Oct. 2002 (invited paper)

206. **Wireless LAN Performance Under Varied Stress Conditions in Vehicular Traffic Scenarios** (with J. Singh, B. Shrinivasan, D. Clawin). Proceedings of the IEEE Vehicular Technology Conference, Vancouver, BC, Canada, pp. 743-747, Oct. 2002.

207. **A Practical Adaptive Algorithm for Power Control in Wireless Networks** (with D. Vengerov, H. Berenji). Proceedings of the 2002 Symposium on Autonomous Intelligent Networks and Systems, University of California at Los Angeles, Los Angeles, CA, May 2002.

208. **Power-Controlled Data Prefetching/Caching in Wireless Data Networks** (with S. Gitzenis). Proceedings of the 20th IEEE Conference on Computer Communications (INFOCOM 2002), New York, NY, vol. 3, pp. 1405-1414, June 2002.

209. **Power Control for Multirate Wireless Networks with Groupwise Serial Multiuser Detection** (with J-W. Kim). Proceedings of the 2001 IEEE Global Communications Conference (GLOBECOM 2001), San Antonio, Texas, vol. 5, pp. 3201-3205, November 2001.

210. **Multi-Modal Dynamic Multiple Access (MDMA) in Wireless Packet Networks** (with S. Kandukuri). Proceedings of the 20th IEEE Conference on Computer Communications (INFOCOM 2001), Anchorage, Alaska, vol. 1, pp. 199-208, April 2001.

211. **Power Controlled Multiple Access (PCMA) in Wireless Communication Networks** (with S. Kandukuri). Proceedings of the 19th IEEE Conference on Computer Communications (INFOCOM 2000), Tel Aviv, Israel, vol. 2, pp. 386-395, March 2000.

212. **Multi-Channel Power Control for Data Traffic in Wireless Networks** (with S. Kandukuri). Proceedings of the 1999 IEEE International Workshop on Mobile Multimedia Communications (MoMuC '99), pp. 83-92, Nov. 1999.

213. **Queueing Networks with Interacting Service Resources** (with M. Armony). Proceedings of the 37th Allerton Conference on Communication, Control and Computing. Monticello, Illinois. Sept. 1999. (invited paper)

214. **Routing in Networks with Random Topologies** (with K. Scott). Proceedings of the 1997 International Communications Conference (ICC '97), vol. 2, pp. 862-866, 1997.

215. **Power Control and Time Division Multiple Access: The TDMA vs. CDMA Question** (with J. Rulnick). Proceedings of the 1997 IEEE Conference on Computer Communications (INFOCOM '97), Kobe, Japan, vol. 2, pp. 635-642, April 1997.

216. **Modeling Communication Links in Percolating Environments** (with N. Alexopoulos). Proceedings of the 1997 IEEE International Conference on Advances in Communications and Control (COMCON '97), pp. 249-261, 1997.

217. **Formation and Maintenance of Self-Organizing Wireless Networks** (K. Scott). Proceedings of the 31st Asilomar Conference on Signals, Systems and Computers, Monterey, CA, vol. 1, pp. 31-35, Nov. 1997.

218. **Algorithmic Speed-up of All Pairs Shortest paths Computation with Reconfigurable Optical Interconnection** (with S.M.P. Yip). Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '96), San Jose, CA, vol. 1, pp. 355-366, August 1996.

219. **Performance Evaluation of Power-Managed Mobile Communication Devices** (with J. Rulnick). Proceedings of the 1996 International Conference on Communications (ICC '96), Dallas, Texas, vol. 3, pp. 1477-1481, June 1996.

220. **Scalable Communication Schemes for Massively Parallel Processing Using Reconfigurable Optical Interconnects** (with S.M.P. Yip). Proceedings of the 1996 International Conference on Parallel & Distributed Systems (ICPADS '96), Tokyo, Japan, pp. 500-507, June 1996.

221. **Best Effort Bandwidth Reservation in High Speed LANs Using Wormhole Routing** (with B. Kwan, P-C. Hu, L. Kleinrock, Hong Xu, Joseph Touch). Proceeding of 5th IEEE International Conference on Computer Communications and Networks (ICCCN '96), Rockville, Maryland, 1996.

222. **Mobile Power Management for Maximum Battery Life in Wireless Communication Networks** (with J. Rulnick). Proceedings of the 1996 IEEE Conference on Computer Communications (INFOCOM '96), San Francisco, CA, vol. 2, pp. 443-450, March 1996.

223. **The Supercomputer Supernet (SSN): A High-Speed Electro-Optical Campus and Metropolitan Network** (with L. Kleinrock, J. Bannister, L. Bergman, J. Cong, E. Gafni, M. Gerla). Proceedings of the 1996 IEEE/ SPIE Conference on Optical Interconnects in Broadband Switching Architectures, San Jose, pp. 22-33, Feb. 1996.

224. **Optimal Server Allocation to Parallel Queues with Randomly Modulated Service Rates** (with G. Michailidis). Proceedings of the Conference on Information Sciences and Systems, Princeton University, Princeton, New Jersey, pp. 692-698, March 1996.

225. **Routing and Channel Assignment for Low Power Transmission in Personal Communication Systems** (with K. Scott). 1996 IEEE International Conference on Universal Personal Communications, vol. 2, pp. 498-502, 1996.

226. **On the Stationary Dynamics of Parallel Queues with Random Server Connectivities** (with G. Michailidis). Proceedings of the 34th IEEE Conference on Decision and Control (CDC '95), New Orleans, LA, vol. 4, pp. 3638-3643, Dec. 1995.

227. **The Self-Organizing Wireless Adaptive Network (SWAN) Protocol for Communication Among Mobile Users** (with K. Scott). Proceedings of the 1995 IEEE Global Telecommunications Conference (GLOBECOM '95), Singapore, vol. 1, pp. 355-359, Nov. 1995.

228. **Channel Probing for Distributed Access Control in Wireless Communication Networks** (with S. Chen, D. Mitra). Proceedings of the 1995 IEEE Global Telecommunications Conference (GLOBECOM '95), Singapore, v. 1, pp.322-326, November 1995.

229. **Radio Link Admission Algorithms for Wireless Networks with Power Control and Active Link Quality Protection** (with S. Chen, G. Pottie). Proceedings of the 1995 IEEE Conference on Computer Communications (INFOCOM '95), Boston, MA, v. 1, pp. 97-104, June 1995.

230. **Entropy Methods for High Speed Communications** (with B. Prabhakar). Proceedings of the 1995 Conference on Information Sciences and Systems, The Johns Hopkins University, Baltimore, Maryland, , pp. 448-453, March 1995.

231. **Adaptive (T1/T2)-Multiplexing Transmission Schemes for Voice/Data Networks** (with A. Nguyen, M. Sherif) Proceedings of the 1995 IEEE International Symposium on Computers and Communications (ISCC '95), pp. 430-435, June 1995.

232. **A Distributed, Mobile, Wireless Infrastructure for Multimedia Applications** (with S. Chen, M. Gerla, J. Tsai). Proceedings of the 1995 WINLAB Conference, Rutgers University, East Brunswick, NJ, March 1995.

233. **Admission Control Schemes for Wireless Communication Networks with Adjustable Transmitter Powers** (with S. Chen, G. Pottie). Proceedings of the 1994 IEEE Conference on Computer Communications (INFOCOM '94), Toronto, Canada, vol. 1, pp. 21-28, May 1994.

234. **On Distributed Power Control for Radio Networks** (with S. Chen, G. Pottie). Proceedings of the 1994 International Communications Conference (ICC '94), New Orleans, LA, vol. 3, pp. 1281-1285, March 1994.

235. **On Power Control with Active Link Quality Protection in Wireless Communication Networks** (with S. Chen and G. Pottie). Proceedings of the 1994 Conference on Information Sciences and Systems, Princeton, NJ, pp. 426-467, March 1994.

236. **The Asymptotics of Traffic Processes in Large Queueing Networks** (with T. Mountford and B. Prabhakar). Proceedings of the 1994 Allerton Conference on Communication, Control and Computing, University of Illinois at Urbana-Champaign, Urbana, Illinois, pp. 563-572, Oct. 1994.

237. **Queueing Dynamics and Throughput of Ring Networks with Spatial Reuse** (with A. Nguyen). Proceedings of the 1993 Allerton Conference on Communication, Control and Computing, University of Illinois at Urbana-Champaign, Urbana, Illinois, pp. 576-587, Oct. 1993 (invited paper).

238. **On Parallel Processing of Random Graph Structured Jobs** (with S. Chen). Proceedings of the 1993 NSF Design and Manufacturing Systems Conference, University of North Carolina at Charlotte, Charlotte, NC, pp. 3-11, Jan. 1993.

239. **Power Control Based Admission Policies in Cellular Radio Networks** (with G. Pottie). Proceedings of the 1992 IEEE Global Telecommunications Conference (GLOBECOM '92), Orlando, Florida, vol. 2, pp. 863-867, Dec. 1992.

240. **Optimal Message Flow on Ring Networks with Spatial Reuse** (with A. Nguyen). Proceedings of the 31st IEEE Conference on Decision and Control (CDC '92), Tuscon, AZ, vol. 2, pp. 2360-2363, Dec. 1992.

241. **On Power Control in High Capacity Cellular Radio Networks** (with G. Pottie). Proceedings of the 1992 WINLAB Conference, Rutgers University, East Brunswick, New Jersey, pp. 239-247, April 1992.

242. **On Parallel Processing in Tandem Queueing Networks with Feedback**. Proceedings of the 1992 NSF Design and Manufacturing Systems Conference, Georgia Institute of Technology, Atlanta, GA,

pp. 495-498, Jan. 1992.

243. **On Tandem Queueing Networks with Feedback** (with K. Wasserman). Proceedings of the 1991 Allerton Conference on Communication, Control, and Computing, University of Illinois at Urbana-Champaign, pp. 310-312, Oct. 1991.

244. **Queueing Theoretic Analysis of a General Class of Concurrent Processing Systems**. Proceedings of the 1991 NSF Design and Manufacturing Systems Conference, The University of Texas at Austin, Texas, pp. 1-5, January 1991.

245. **Maximal Throughput for Stability of a Class of Parallel Processing Systems** (with J. Walrand). Proceedings of the 29th IEEE Conference on Decision and Control (CDC '90), Honolulu, Hawaii, vol. 1, pp. 161-166, Dec. 1990.

246. **Flow Analysis of Stochastic Marked Graphs** (with F. Baccelli, J. Walrand). Proceedings of the 28th IEEE Conference on Decision and Control (CDC '89), Tampa, Florida, vol. 2, pp. 1528-1531, Dec. 1989.

247. **On Bifurcated Shortest Path Routing in Communication Networks** (with G. Stassinopoulos). Proceedings of the 1984 IEEE European Conference on Control and Communications, Patras, pp. 604-609, July 1984.

REPORTS:

248. **ARQ: A Multiclass Admission Control Protocol for Heterogeneous Datacenters** (with C. Delimitrou, C. Kozyrakis). Stanford Technical Multiscale Architecture & Systems Lab. Stanford Technical Report, TR-2013-002, Jan. 2013.

249. **Adaptive Data-Aware Utility-Based Scheduling in Resource-Constrained Systems** (with D. Vengerov, L. Mastroleon, D. Murhpy). Sun Microsystems Labs Technical Report SMLI TR-2007-164, April 2007.

250. **Architecting Novel Information Services and Markets,** Detecon Management Report, 2006.

251. **Wireless Packet Scheduling with Deadlines.** Stanford Technical Report, 2005

252. **Projective Processing Schedules in Queueing Structures: Applications to Packet Scheduling in Communication Networks Switches.** Stanford Technical Report. 2002.

253. **Network Security and Infrastructure Assurance – Architectural and Performance Engineering Issues**. Stanford Technical Report, 2004

254. **Noninvasive Channel Probing for Distributed Admission Control and Channel Allocation in Wireless Networks** (with J-W. Kim, D. Mitra, S. Chen). Stanford Technical Report, 2002

255. **Power-Controlled Data Prefetching and Caching in Wireless Mobile Networking** (with S. Gitzenis). Stanford Technical Report, 2002.

256. **Queueing Networks with Random Link Topologies; Stationary Dynamics of Maximal Throughput Schedules** (with G. Michailidis). Stanford Technical Report, 2001.

257. **On Stationary Multi-Processor Queues with Threshold-Based Server Activation** (with J. Rulnick). Technical Report: NetLab 1997-09-01, Stanford University, 1997.

258. **Convergence of Departures in Tandem Networks of ./GI/Infinity Queues** (with T. Mountford, B. Prabhakar). Technical Report: HPL-BRIMS-96-15, BRIMS, HP Labs, UK, 1996.

259. **Processor Migration in Distributed Queueing: An Analysis of the two Processor Case** (with K. Scott). Technical Report UCLA-ENG-95-138, School of Engineering and Applied Science, UCLA, 1995.

260. **Multicopy Parallel Processing Networks: Throughput Analysis** (with K. Scott). Technical Report UCLA-ENG-95-137, School of Engineering and Applied Science, UCLA, 1995.

261. **Critical Phenomena in Massively Parallel Processing Networks: Throughput Analysis - Initial Results** (with A. Zarkesh). Technical Report UCLA-ENG-95-141, School of Engineering and Applied Science, UCLA, 1995.

262. **A Fully Distributed Algorithm for the Creation and Maintenance of Wireless Networks of Mobile Users** (with K. Scott). Technical Report UCLA-ENG-95-112, Department of Electrical Engineering, University of California at Los Angeles, 1995

263. **Probing the Collective Behavior of Massively Parallel Processing Systems by the Renormalization Group Method** (with A. Zarkesh). Technical Report UCLA-ENG-95-140, School of Engineering and Applied Science, UCLA, 1995.

264. **Performance of a Switch State Dependent Timeout Scheme in a Wormhole Switching LAN** (with B. Kwan). UCLA Technical Report UCLA- ENG-95-121, School of Engineering and Applied Science, UCLA, 1995.

265. **Branching Processes in Rollback Synchronization of Parallel Processing Systems: Structural Results** (with J. Rulnick). Technical Report UCLA-ENG-95-136, School of Engineering and Applied Science, UCLA, 1995.

266. **Optimal Choice of Queue Length Thresholds for Processor Activation in the M/M/K Queue** (with J.M. Rulnick). Technical Report UCLA-ENG-95-134, School of Engineering and Applied Science, UCLA, 1995.

267. **Partial Worm Retraction Routing Scheme for Wormhole Switching** LANs (with B. Kwan). UCLA Technical Report UCLA ENG-94-108, November 1994.

268. **Admission Control and Routing Protocols in Stationless Multi-Hop Radio Networks** (with K. Scott). Technical Report UCLA-ENG-94-106, School of Engineering and Applied Science, U.C. Los Angeles, 1994.

269. **Optimistic (Partial Retraction) Routing Schemes for Wormhole Switching LANs** (with B. Kwan). Technical Report UCLA-ENG-94-108, School of Engineering and Applied Science, U.C. Los Angeles, 1994.

270. **Radio Link Admission Algorithms for Wireless Networks with Power Control and Active Link Quality Protection** (with S. Chen, G. Pottie). Technical Report UCLA-ENG-94-25. School of Engineering and Applied Science, U.C. Los Angeles, 1994.

271. **Distributed Dynamic Power Control Schemes in High Capacity Wireless Radio Networks** (with S. Chen, G. Pottie). Technical Report UCLA-ENG-93-57, School of Engineering and Applied Science, U.C. Los Angeles, 1993.

272. **A Localized Power and Admission Control Scheme Adapted to Varying Channel Conditions for High Capacity Wireless Radio Networks** (with S. Chen, G. Pottie). Technical Report UCLA-ENG-93-56, School of Engineering and Applied Science, U.C. Los Angeles, 1993.

273. **Admission Control Schemes for Guaranteed Link Quality in High Capacity Cellular Radio Networks** (with S. Chen, G. Pottie). Technical Report UCLA-ENG-93-55, School of Engineering and Applied Science, U.C. Los Angeles, 1993.

274. **On Generalized Multiserver Queues** (with J. Walrand). Technical Report UCLA-ENG-9-24-90, School of Engineering and Applied Science, U.C. Los Angeles, 1990.

275. **Recent Results on the Asymptotic Behavior of Concurrent Processing Systems Under Stationary and Ergodic Inputs** (with J. Walrand). Technical Report UCB/ERL M88/78, Electronics Research Laboratory, U.C. Berkeley, 1988.

276. **An Alternative Approach to the Asymptotic Makespan of Serially Executed Multitasked Jobs** (with J. Walrand). Technical Report UCB/ERL M88/79, Electronics Research Laboratory, U.C. Berkeley, 1988.

TALKS:

1. **Invited Seminar – A Risk Management Approach to Cyber-Security**, Operations Research Program, School of Engineering, University of Texas at Austin, March 2017.

2. **Predicting Pediatric Surgical Case Durations** (published abstract with N. Master, D. Scheinker). INFORMS International Meeting, Hawaii, 2016.

3. **Myopic Scheduling of Jobs with Decaying Value with Applications**, (abstract with N. Master, C.W. Chan) INFORMS International Meeting, Hawaii, 2016. (content essentially the same as of talk below.)

4. **Games on Influence Networks: Equilibria, Free Riding and Dynamics** (with Z. Zhou, P. Glynn). INFORMS Annual Meeting, Nashville, Tennessee, 2016.

5. **Myopic Scheduling of Jobs with Decaying Value with Applications in Patient Scheduling** (published abstract with N. Master, C.W. Chan). INFORMS Annual Meeting, Philadelphia, PA, 2015.

6. **Distinguished Invited Seminar: Power and Throughput Issues in Next-Generation Packet Switches**, National Technical University of Athens, Greece, Feb. 2009.

7. **Distinguished Invited Seminar: Wireless Multimedia Computing**, DoCoMo Research Labs, Palo Alto, Feb. 2009.

8. **Plenary Talk: Architecting Novel Information Services and Markets**, 4th International IMS Workshop, Fraunhofer Institute for Open Communication Systems – FOKUS, Berlin, Germany, Nov. 2008.

9. **Video Streaming over Wireless Networks – Control Architectures and Performance Engineering**, NEC Research Labs, Tokyo, July 2008.

10. **Power Management in Networking and Computing,** Hitachi Labs, Tokyo, July 2008.

11. **Power Management of Packet Switches**, Sun Labs, Menlo Park, April 2008.

12. **Power and Throughput Issues in Next-Generation Packet Switches**, UC Irvine, February 2008.

13. Invited Distinguished Lectures: 1) **Robust Maximal Throughput Schedules in Processing Systems**, 2) **Transmitter Power Control in Wireless Multimedia Networking**, Advanced Network Sciences Institute, University of California, San Diego, August 2008.

14. **Autonomous Admission (and Power) Control in Wireless Networking**, Technical University of Berlin, December 2007.

15. **Video Streaming over Wireless Networks – Control Architectures and Performance Engineering**, Samsung Labs, Santa Clara, December 2008.

16. **Geometry of Resource Allocation and Load Balancing in Computing/Networking**, Stanford OR Seminar, November 2007.

17. **Throughput and Power Issues in Packet Switching**, University of Sydney, Australia, August 2007.

18. **Video Streaming over Wireless Networks – Control Architectures and Performance Engineering**, Bogazici University, Istanbul, Turkey, August 2007.

19. **Outsourcing in Computing and Data Storage Services**, Deutsche Telekom Laboratories, Berlin, August 2007.

20. **Dynamic Control of Packet Switches**, Sun Labs, Menlo Park, February 2007.

21. **Geometry of Resource Allocation and Load Balancing in Computing/Networking**, Deutsche Telekom Laboratories, Berlin, January 2007.

22. **Network Control Issues in Video Transmission and Distribution,** Deutsche Telekom Labs, August 2006.

23. **Power Control in Wireless Networks**, IBM Research Labs, Hawthorne, July 2006.

24. **Scheduling and Power Management Issues in Data Centers and High-Performance Computing**, Sun Labs, Menlo Park, May 2006.

25. **Video Streaming over Wireless Networks – Control Architectures and Performance Engineering,** Athens Information Technology Institute, May 2006.

26. **Power Control in Wireless Networks**, ENS, Paris.

27. **Video Streaming over Wireless Networks – Control Architectures and Performance Engineering,** Technical University of Athens, December 2005.

28. **Geometry of Resource Allocation and Load Balancing in Computing/Networking**, Athens University of Economics and Business, November 2005.

29. **Power Control in Wireless Networks**, Sprint Labs, Burlingame, December 2004.

30. **'Geometry' of Packet Scheduling in Communication Switches**, MIT LIDS Seminar, November 2004.

31. **Power(Rate)/Playout-Controlled Video over Wireless**, France Telecom Research Labs, Paris, June 2004.

32. **Next-Generation Wireless Data/Media Networks**, Athens Information Technology Institute, February 2004.

33. **Projective Packet Scheduling in Switching Systems**, France Telecom Research Labs, Paris, February 2004.

34. **Next-Generation Wireless Data/Media Networks**, France Telecom Research Labs, February 2004.

35. **Geometry of Packet Scheduling Algorithms for Input-Queued Communication Switches.** INFORMS Conference, San Jose, CA, Nov. 2002.

36. **Adaptive Admission Control for Multiclass Web Servers** (with G. Paleologo). INFORMS Telecommunications Conference, Boca Raton, FL, March 2002.

37. **Power-Controlled Data (Pre)Fetching and Caching in Wireless Networking** (with S. Gitzenis). INFORMS Telecommunications Conference, FL, March 2002.

38. **On Stability of Queueing Networks with Job Deadlines** (with A. Ward). INFORMS Applied Probability Conference, New York City, NY, July 2001.

39. **Throughput Maximizing Properties of Statistical Multiplexing in Stationary Ergodic Networks with Feedback** (with C. Westphal). INFORMS Applied Probability Conference, New York City, NY, July 2001.

40. **Predictive Caching Techniques in Wireless Networks** (with A. Ward). INFORMS Telecommunications Conference, Boca Raton, FL, March 2000.

41. **Dynamic Cone Policies for Skilled Based Routing in Call Centers** (with M. Armony). INFORMS Telecommunications Conference, Boca Raton, FL, March 2000.

42. **Dynamic Cone Policies for Queueing Networks with Interfering Resources** (with M. Armony). INFORMS Conference, Philadelphia, PA, November 1999.

43. **Stability and Performance Issues of a General Heterogeneous Parallel Processing System** (with G. Michailidis, K. Wasserman, S. Akin). INFORMS Applied Probability Conference, Ulm, Germany, July 1999.

44. **Bandwidth Allocation in Wireless Networks** (with G. Michailidis). INFORMS Conference, Seattle, WA, October 1998.

45. **Stability and Large Buffer Asymptotics of a General Parallel Processing System** (with G. Michailidis, K. Wasserman, S. Akin). Conference of the Canadian Operational Research Society, Windsor, Ontario, June 1999.

46. **Dynamic Cone Policies for Queueing Networks with Interfering Resources** (with M. Armony). INFORMS Conference, Philadelphia, PA, November 1999.

47. **Power Control and QoS in Wireless Communication Networks** (with M. Armony). INFORMS Conference, Cincinnati, OH, May 1999.

48. **Scheduling Algorithms for High Performance Switches** (with S. Yip). INFORMS Conference, Seattle, WA, Oct. 1998.

49. **Bandwidth Allocation in Wireless Networks** (with G. Michailidis). INFORMS Conferences., Seattle, WA, October 1998.

50. **Queueing in Interfering Resource Environments** (with M. Armony). INFORMS Conference, Tel Aviv, Israel, June 1998.

51. **Allocation of Interdependent Resources in Queueing Systems with Applications to Wireless Networks** (with M. Armony). INFORMS Telecommunications Conference, Boca Raton, FL, March 1998.

52. **On Performance and Limitations of Ultra Large Scale Digital Interconnects with Reconfigurable Topologies** (with S. Yip). IEEE/LEOS Conference on Interconnection of High Speed Digital Systems, Santa Fe, NM, May 1997.

53. **Queueing in Random Environments** (with G. Michailidis). INFORMS Applied Probability Conference, Boston, MA, June 1997.

54. **Network Flows in Random Environments** (with G. Michailidis). INFORMS Applied Probability Conference, Atlanta, GA, June 1995.

55. **On Queueing Networks with Random Link Topologies** (with G. Michailidis). INFORMS Telecommunications Conference, Boca Raton, FL, March 1995.

56. **Power Control Issues in Wireless Networks** (with S. Chen, G. Pottie). INFORMS Telecommunications Conference, Boca Raton, FL, March 1995.

57. **The Entropy and Delay of Traffic Processes in ATM Networks** (with B. Prabhakar). 1995 IEEE Information Theory Workshop on Multiple Access and Queueing, St. Louis, Missouri, April 1995.

58. **Scalable Supercomputer Network** (with L. Kleinrock, N. Bambos, L. Bergman, J. Cong, E. Gafni, M. Gerla). 1994 IEEE Gigabit Networking Workshop, Toronto, Canada, June 1994.

59. **The Supercomputer Supernet Hardware Design** (with L. Kleinrock, N. Bambos, L. Bergman, J. Cong, E. Gafni, M. Gerla). DARPA Networking Systems Meeting, Santa Fe, NM, September 1994.

60. **Dynamics of Parallel Processing: A Non-Causal Queue** (with J. Rulnick). Annual meeting of the Society of Industrial and Applied Mathematics (SIAM), San Diego, CA, July 1994.

61. **On the Synchronization of Poisson Processes and Queueing Networks with Synchronization Nodes** (with B. Prabhakar and T. Mountford). Fifth Workshop on Stochastic Analysis and Related Fields, Silivri,

Turkey, July 1994.

62. **On Flow Control in Ring Structured Local Area Networks with Spatial Reuse** (with A. Nguyen). INFORMS Telecommunications Conference, Boca Raton, Florida, March 1992.

63. **Stable Policies for Processing Systems with Precedence Constraints**. ORSA/TIMS Joint National Meeting, Anaheim, California, November 1991.

64. **On Stability of Parallel Processing Systems** (J. Walrand). SIAM Conference on Applied Probability in Science and Engineering, New Orleans, LA, March 1990.

65. **State-Dependent Queues and Queueing Networks** (with J. Walrand). International Conference on Stochastic Programming, Ann Arbor, MI, August 1989.

66. **Recent Developments in Power Control and their Impact on Next-Generation Dense Wireless Network** (invited distinguished talk). Wireless/Sensor Network Track of the 2002 IEEE Wireless Communications and Networking Conference (WCNC 2002), Orlando, Florida, March 2002.

67. **Power Control and Management Issues in Next-Generation Wireless Networks** (invited keynote talk). 2001 IEEE Symposium on Ad Hoc Wireless Networks (SAWN 2001) of the 2001 IEEE Global Communications Conference (GLOBECOM 2001), San Antonio, Texas, November 2001.

68. **Queueing Models and Problems in Wireless Network Contro**l (invited plenary talk). Stochastic Networks Conference, Madison, WI, June 2000.

69. **Power Controlled Multiple Access (PCMA) in Wireless Networking** (invited talk). IEEE Computer Communications Workshop, Estes Park, Colorado, November 1999.

70. **Queueing Networks with Interdependent Resources** (invited plenary talk). Oberwolfach Workshop on Applied Probability,, Mathematisches Forschungsinstitut, Oberwolfach, Germany, December 1998.

71. **Power Control Issues in Wireless Communication Networks** (invited plenary talk). Workshop on System and Control Issues in Communication Networks, National Science Foundation and Army Research Office, Airlie Center, Airlie, Virginia, August 1996.

72. **On Queueing Networks with Random Link Topologies: Bandwidth Allocation in Wireless Communication Networks** (invited plenary talk). BRIMS Workshop on Mathematical Trends in Communication Networks, Hewlett-Packard Labs, Bristol, UK, June 1996.

73. **Power Control with Active Link Protection in Wireless Networking** (invited plenary talk). EPFL Workshop on Computer Networking, EPFL, Switzerland, Summer 1996.

74. **Power Control Algorithms in High Performance Wireless Networks** (invited talk). Networking Seminar, IBM Watson Research Center, NY, July 2001.

75. **Queueing Networks in Random Environments** (invited talk). Networking Seminar, Dept. of EECS, U.C. Berkeley, March 1998.

76. **On Queueing Networks with Randomly Modulated Service Rates** (invited talk). Research Colloquium, Center for Operations Research, MIT, September 1997.

77. **High-Performance Wireless Networking**. Networking Seminar, Dept. of EECS, U.C. Berkeley, Nov. 1996

ISSUED PATENTS

Nicholas Bambos

October 22, 2018

1. **US 6,711,280 B2** 2004/2001 **–** *Method and Apparatus for Intelligent Ranging via Image Subtraction*.

2. **US 7,317,730 B1** 2008/2001 **–** *Queueing Architecture and Load Balancing for Parallel Packet Processing in Communication Networks*.

3. **US 7,634,287 B1** 2009/2001 – *Power Controlled Multiple Access (PCMA) in Wireless Communication Networks*.

4. **US 8,320,269 B2** 2012/2009 **–** *Scalable Delay-Power Control Algorithm for Bandwidth Sharing in Wireless Networks*.

5. **US 8,423,070 B2** 2013/2004 **–** *Method and System to Model TCP Throughput, Assess Power Control Measures, and Compensate for Fading and Path Loss, for Highly Mobile Broadband Systems*.

6. **US 8,488,500 B2** 2013/2008, **US 9,454,209 B2** 2016/2008, **EP 2,272,287 B1** 2017/2008 (European counterpart with English, French and German versions), **JP 5,997,794 B2** 2016/2008 (Japanese counterpart) – *Power Management of Networked Devices* – (and pending application **US 15/909,619**, filed on 2018-03-01, which is a continuation of pending application **US 15/276,573**, filed on 2016-09-26, which is a continuation of application **US 13/942,586**, filed on 2013-07-15 (now patent US 9,457,209), which is a continuation of application **US 12/114,721**, filed on 2008-05-02 (now patent US 8,488,500); and pending application **IN 6987/CHENP/2010** in India).

7. **US 9,614,958 B2** 2017/2007, **EP 2,132,921 B1** 2018/2007 (European counterpart) – *Predictive Computer Network Services Provisioning for Mobile Users*.

--- CONFIDENTIAL ---

Please note that most engagements identified below are governed by confidentiality agreements.

December 19, 2022

A) EXPERT WITNESS ENGAGEMENTS OF NICHOLAS BAMBOS

I, Nicholas Bambos, have had the following temporary expert witness engagements in the litigation cases listed below:

1. (2002-2003) Technical consultant and expert witness (*deposed*) for *Altima Communications, Inc.* in the case "Level One Communications, Inc. v. Altima Communications, Inc." Case No. CIV. S-99-2488 GEB (GGH), US District Court, Eastern District of California

2. (2004-2005) Technical consultant and expert witness (*deposed*) for *Microsoft Corporation* in the litigation case "Lextron Systems, Inc. vs. Microsoft Corporation" Case No. CV 04-00588-VRW, US District Court, Northern District of California, San Francisco Division

3. (2004-2005) Technical consultant and expert witness (*not* deposed) for *Quova, Inc.* in the case "Quova, Inc. v. Digital Envoy, Inc." Case No. C04 02674JL, US District Court, Northern District of California, San Jose Division.

4. (2004-2005) Technical consultant and expert witness (*deposed*) for *American Honda Motor Co., Inc.* in the litigation case "Donner, Inc. (Plaintiff and Counter-Defendant) v. American Honda Motor Co. Inc. and McDavid Plano-Acra, L.P. and The Beaumont Company d/b/a Classic Acura, (Defendants and Counter-Plaintiffs) v. Irah H. Donner (Counter-Defendant)" Civil Action File No. 5:03-CV-253 (DF), US District Court, Eastern District of Texas, Texarkana Division.

5. (2005-06) Technical consultant and expert witness (*deposed*) for *Pulse Engineering Inc. and Full Rise Electronics Co.* in the litigation case "Regal Electronics, Inc. v. Pulse Engineering, Inc., Fahrner-Miller Associates, Inc., Full Rise Electronics Co. Ltd., Max Lion Electronics, Inc., Bell Fuse, Inc., Stewart Connector Systems, Inc." Civil Case No**:** 03-01296 JW, US District Court, Northern District of California, San Jose Division.

6. (2005-2008) Technical consultant and expert witness (*not* deposed) for *Realtek Semiconductor Corporation* in the litigation case "3COM Corporation v. D-Link Systems, Inc. and Realtek Semiconductor Corporation" Case No. CV-03-2177 VRW BZ, US District Court, Northern District of California, San Francisco Division.

7. (2006-2007) Technical consultant and expert witness (*deposed*) for *Tekelec, Inc.* in the litigation case "Bouygues Telecom, S.A. vs. Tekelec, Inc." Case No. 4:05 CV 78-FL3, US District Court, Eastern District of North Carolina.

8. (2006-2008) Technical consultant and expert witness (*deposed*) for *Nissan North America, Inc.* in the litigation case "Mobile Micromedia Solutions LLC vs. Nissan North America, Inc." Civil Action File No. 5:05-CV-230, US District Court, Eastern District of Texas, Texarkana Division.

9. (2006-2013) Technical consultant and expert witness (*not* deposed) for *J2 Global Communications, Inc.* and *Catch Curve, Inc.* in the litigation cases (all related) “J2 Global Communications Inc. v. CallWave, Inc.” Civil Action No. CVO4-706D8 DP (AJWx); and “J2 Global Communications Inc. v. Venali, Inc.” Civil Action No. CVO4-ll72DDP (AJWx); and “Catch Curve, Inc. (Plaintiff/Counterclaim-Defendant) v. Venali, Inc. (Defendant/Counterclaimant)” Civil Action No. 05-4820 DDP (AJWx), US District Court, Central District of California; and “ J2 Global Communications, Inc. and Advanced Messaging Technologies, Inc. v. Protus IP Solutions, Inc.” Civil Action No. CV05-5610 DDP (AJWx); and “COA Network, Inc. v. J2 Global Communications, Inc.”; and “J2 Global Communications, Inc. and Advanced Messaging Technologies, Inc. v. Captaris, Inc. and Open Text Corp.” Civil Action No. CV09-4150 DDP (AJWx); and J2 Global Communications, Inc. and Advanced Messaging Technologies, Inc. v. Packetel, Inc.” Civil Action No. CV09-3240 DDP (AJWx); and “J2 Global Communications, Inc. and Advanced Messaging Technologies, Inc. v. Easylink Services International Corporation” Civil Action No. CV 09-4189 DDP (AJWx); and “J2 Global Communications, Inc. and Advanced Messaging Technologies, Inc. v. Venali, Inc.” CV04-1172 DDP (AJWx); and the related case in Canada “J2 Global Comunications, Inc. and Catch Curve Inc. v. Protus IP Solutions Inc.” Canadian Federal Court Action No. T-139-06.

10. (2007) Technical consultant and expert witness (*not* deposed) for *Netgear* in the litigation case “Serconet, Ltd. (Plaintiff) v. Netgear, Inc. (Defendant/Third-Party Plaintiff) v. Cameo Communications, Inc., Delta Networks, Inc., Hon Hai Precision Industry Co., Ltd., Sercomm Corp., and Z-Com, Inc. (Third Party Defendants)” Civil Action No. C-06-04646 PJ, US District Court, Northern District of California, San Francisco Division.

11. (2008) Technical consultant and expert witness (*not* deposed) for *Quantum Corporation* in the litigation case “Quantum Corporation v. Riverbed Technology, Inc.” Case No. C-08-0927-WHA, US District Court, Northern District of California.

12. (2008) Technical consultant and expert witness (*deposed*) for Vero Systems (Teoco) in the litigation case “Telarix, Inc. vs. Vero Systems, Inc.” Eastern District of Virginia Case No. 1-08-cv-587 (CMH)

13. (2009-2010) Technical consultant and expert witness (*not* deposed) for ActiveIdentity Corporation in the litigation case “ActiveIdentity vs. Intercede Group PLC and Intercede Ltd.” Northern District of California, Case No. C-08-4577 VRW.

14. (2010-2011) Technical consultant and expert witness (*not* deposed) for Perfect 10 Satellite Distributing Company in the litigation case “John Mezzalingua Associates, Inc. vs. Pace Electronics, Inc., Perfect 10 Antenna Company and DS Engineering, LLC” Case Number: 0.10-cv-00064-MJD-JJG.

15. (2010) Technical consultant and expert witness (*not* deposed) for Belkin International, Inc., Cisco-Linksys L.L.C., D-Link Systems, Inc., Netgear, Inc., and SMC Networks, Inc. in the litigation case “OptimumPath, L.L.C. v. Belkin Int’l, Inc., et al” Case No. 4:09-CV-1398-CW (N.D. Cal)

16. (2011) Technical consultant and expert witness (*not* deposed) for Infoblox, Inc. in the litigation case “Infoblox v. BlueCat” Case Number CV 10-01962 JVS.

17. (2011-2012) Technical consultant and expert witness (*deposed*) for Twitter, Inc. in the litigation case “Copper Notification, Inc., v. Twitter, Inc., Everbridge, Inc., Rave Wireless, Inc., Federal Signal Corp” Civil Action No. 09-865-LPS.

18. (2011-2014) Technical consultant and expert witness (*deposed*) for Barnes & Noble, Inc. in the litigation case “Barnes & Noble, Inc. v. LSI Corp.” No. 3:11-CV-2709-EMC, U.S.D.C. N.D. Cal.

19. (2011-2013) Technical consultant and expert witness (*not* deposed) for Google, Inc. in the litigation case "MasterObjects, Inc. v. Coogle, Inc." Case No. CV 11-01054 PJH, Northern District of California, Oakland Division.

20. (2012-2013) Technical consultant and expert witness (*not* deposed) for Money Suite Company in the litigation case "The Money Suite Company v. Insurance Answer Center, LLC, et al," USDC Central District of California – Case No. SACV 11-1847-AG (JPRx)

21. (2012-2013) Technical consultant and expert witness (*deposed*) for Brilliant Telecommunications / Juniper Networks in the litigation case "Alcatel Lucent Technologies USA Inc. v. Brilliant Telecommunications, Inc., et al," Case No.: 1-11-CV-206910, Superior Court of California, County of Santa Clara.

22. (2012-2013) Technical consultant and expert witness (*deposed & testified at ITC trial*) for HTC Corporation / HTC America Inc. (Respondents) and for Google, Inc. (Intervener) in "The Matter of Certain Electronic Devices, Including Mobile Phones, Tablet Computers, and Components Thereof, ITC Inv. No. 337-TA-847." (Complainants: Nokia Corp., Nokia Inc., Intellisync Corp.) Worked on three patents, two of which were dropped by Complainants and the Judge found no infringement on the third.

23. (2013) Technical consultant and expert witness (*not* deposed) for HTC Corporation / HTC America, Inc. in the litigation case "Smartphone Technologies vs. AT&T Mobility LLC, HTC Corporation, HTC America, Inc." Case No. 6:11-CV-561 (E.D. Tex.).

24. (2013) Technical consultant and expert witness (*deposed*) for Aeritas, LLC in the litigation cases: 1) Aeritas, LLC v. Alaska Air Group, Inc., Case No. 11-967-SLR (D. Del.); 2) Aeritas, LLC v. Continental Airlines, Inc., Case No. 11-968-SLR (D. Del.); 3) Aeritas, LLC v. Delta Airlines, Inc., Case No. 11-969-SLR (D. Del.); 4) Aeritas, LLC v. United Airlines, Inc., Case No. 11-970-SLR (D. Del.); 5) Aeritas, LLC v. US Airways, Inc., Case No. 11-1267-SLR (D. Del.); and 6) Aeritas, LLC v. Virgin America, Inc., Case No. 12-1492-SLR (D. Del.).

25. (2013-2014) Technical consultant and expert witness (*deposed*) for IpLearn, LLC in the litigation case "IpLearn, LLC v. Blackboard Inc." Case No. 1:11-CV-00876-RGA (D. Del.).

26. (2013-2015) Technical consultant and expert witness (*not* deposed) for Infoblox, Inc. in the litigation case "Versata Software, Inc., et al v. Infoblox, Inc." Case No. 1:13-cv-00678-LPS (D. Del.)

27. (2014-2015) Technical consultant and expert witness (*not* deposed) for Kyocera Corp. and Kyocera Communications Inc. in the litigation case "Adaptix, Inc. v. Kyocera Communications Inc." Case No(s) 5:14-cv-02894-PSG (and -02895-PSG), Northern District of California, and related IPR case(s).

28. (2015-2016) Technical consultant and expert witness (*deposed*) for IBM Corporation in the litigation case "Twin Peaks Software Inc. v. IBM Corporation" Case No: 14-CV-03933-JST.

29. (2016-2019) Technical consultant and expert witness (*deposed*) for Samsung in the litigation case "Huawei Technologies, Co. Ltd. Et al v. Samsung Electronics Co. Ltd. Et al." Case No: 3:16-CV-02787 N.D. Cal.

30. (2016-2017) Technical consultant and expert witness (*not deposed*) for Bunjie, Inc. in the litigation case(s) "Bunjie, Inc. v. Acceleration Bay, LLC." Case No: IPR 2017-01600 (U.S. Patent No. 6,910,069) and Case No: IPR 2017-01601 (U.S. Patent No. 6,829,634).

31. (2016-2018) Technical consultant and expert witness (*not deposed*) for BlackBerry in the litigation case "BlackBerry Limited v. Nokia Corporation, Nokia Solutions and Networks Oy, and Nokia of America Corporation," Case No. 1:17-CV-00155 (RGA), United States District Court for the District of Delaware. (Also provided consultation services for related IPRs, but did not participate as a disclosed expert witness).

32. (2017) Technical consultant and expert witness (*not deposed*) for Amazon.com and Amazon Web Services, Inc. in IPR proceedings related to the litigation case "Broadcom Corporation, et al. v. Amazon.com, Inc. et al." Case No: 8:16-cv-01774-JVS-JCG (C.D. Cal.)

33. (2017-2019) Technical consultant and expert witness (*not deposed*) for Wiko SAS in the patent litigation case "Koninklijke Philips N.V. vs. Wiko SAS," Case No. 200.219.487/01, Court of Appeal of The Hague in the Netherlands; and technical consultant and expert witness (*not deposed*) for AsusTek Computer Inc., Asus Europe B.V. and Asus Holland B.V. in the related patent litigation case "Koninklijke Philips N.V. vs. AsusTek Computer Inc., Asus Europe B.V. and Asus Holland B.V.," Case No. 200.221.250/01, Court of Appeal of The Hague in the Netherlands.

34. (2018-2019) Technical consultant and expert witness (*not deposed*) for Juniper Networks, Inc. in the patent litigation case "Parity Networks, LLC v. Juniper Networks, Inc.," Case No. 6:17-CV-00495-RWS-KNM, United States District Court for the Eastern District of Texas.

35. (2019) Technical consultant and expert witness (*not deposed*) for Metaswitch Networks Ltd. and Metaswitch Networks Corporation in the patent litigation case "Sonus Networks, Inc. (d/b/a Ribbon Communications Operating Company) v. Metaswitch Networks Ltd. and Metaswitch Networks Corporation (counterclaim case)," Civil Action Nos. 2:18-CV-57-RWS, Eastern District of Texas.

36. (2020 -2021) Technical consultant and expert witness (*not deposed*) for Dell Technologies Inc., Dell Inc., and EMC Corp. in the patent litigation case "WSOU Investments, LLC v. Dell Technologies Inc. et al., No 6:20-cv-477 (W.D. Tex.).

37. (2022 -) Technical consultant and expert witness (*not deposed*) for Ericsson Inc., AT&T Corp., AT&T Communications LLC., AT&T Mobility, AT&T Mobility II LLC, AT&T Services Inc. in the patent litigation case "K. Mizra LLC v. AT&T Corp. et al., 2:21-cv-00241 (E.D. Tex.) (Lead Case).

38. (2022 -) Technical consultant and expert witness (*not deposed*) for Dell in the patent litigation case Corrigent Corp. v. Dell Technologies Inc. and Dell Inc., Case No. 22-cv-496 (D. Del.); and also for Cisco in the related patent litigation case Corrigent Corp. v. Cisco Systems, Inc. Case No. 22-cv-396 (W.D. Tex.)

B) OTHER CONSULTING, ETC. ENGAGEMENTS OF NICHOLAS BAMBOS

Additionally, I have had the following startup-founding, technical advising, consulting, and other engagements (with compensation, unless otherwise specified)) in the past.

1. (1994-1996) Research visitor to the AT&T Bell Labs Mathematics Center for two weeks in the summer of 1994 and two weeks in the summer of 1996.

2. (1996-present) Research proposal review panelist at the National Science Foundation.

3. (1998) Research consultant for Sun Microsystems for two weeks in 1998.

4. (1999-2001) Co-founder of "AON Networks" (optical networking technology startup).

5. (2001-2003) Member of the Technical Advisory Board of ChainCast (peer-to-peer audio streaming technology startup; now StreamAudio).

6. (2001-2003) Member of the Technical Advisory Board and consultant for Greenfield Networks (packet switching technology startup; eventually acquired by Cisco).

7. (2001-2004) Advisor at ITU Ventures, Los Angeles, CA (technology investment firm).

8. (2004-2004) Member (uncompensated) of the Advisory Board of Detecon USA (consulting branch of Deutsche Telekom).

9. (2008-2009) Co-founder of Dhaani (initially Verde) Systems (technology startup focusing on power management of desktop computers).

10. (2008) Presentation for Cyber Ark International, Inc. on "science and technology innovation in the Silicon Valley" to Chinese delegation in Oct. 2008.

11. (2010 -) Member (uncompensated) of the Scientific Committee, Laboratory for Information, Networking and Communication Sciences (LINCS), Paris, France.

12. (2012) Consulting for Quinn-Emanuel on patent analysis in Oct. & Nov. of 2012.

13. (2013) Consulting for Kilpatrick-Townsend-Stockton on prior art for defendant in Sept. & Oct. 2013, regarding multimedia messaging services (MMS) technology.

14. (2015) Consulting for Quinn-Emanuel on patent analysis in Jan. to May of 2015.

15. (2015) Consulting for Kirkland-Ellis on patent analysis in April 2015, regarding network resource management technology.

16. (2015) Consulting for Winston-Strawn on prior art for defendant in the period May-September 2015.

17. (2016) Consulting for Quinn-Emanuel on patent litigation in January 2016 (engaged, but performed no work and received no compensation).

18. (2018 - 2020) Member (uncompensated) of the Advisory Board, German-Turkish Advanced Research Center (GT-ARC) for Information and Communication Technologies, Berlin, Germany.

19. (2019 - 2021) Consulting for Desmarais LLP on a patent litigation matter beginning September 2019, regarding wireless network communications.

20. (2021 - 2022) Consulting for an international law firm on a litigation matter beginning February 2021, regarding networking technology.

Appendix B

APPENDIX B: CHALLENGED CLAIM LISTING

No.	Limitation
1[pre]	A method for communication, comprising:
1[a]	configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;
1[b]	configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards;
1[c]	providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;
1[d]	receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;
1[e]	conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;
1[f]	if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;
1[g]	checking said MAC source address of the data packet against records in said FDB of said first line card; and
1[h]	if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card of said plurality of member line cards.

No.	Limitation
2	The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
3	The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.
4	The method according to claim 3, and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
5[pre]	The method according to claim 4, and comprising
5[a]	associating a respective aging time with each of the records;
5[b]	refreshing the records in the FDB responsively to further packets transmitted by the line cards; and
5[c]	removing the records from the respective FDB if the records are not refreshed within the respective aging time.
6	The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via a switching core of the network node to at least the second line card
7	The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.

No.	Limitation
8	The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances
9	The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.
10[pre]	The method according to claim 1, and comprising:
10[a]	conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;
10[b]	checking the further MAC source address against the records in the FDB of the second line card; and
10[c]	responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.
11[pre]	A node for network communication, comprising:
11[a]	a switching core;
11[b]	a plurality of ports;
11[c]	a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards,

No.	Limitation
11[d]	wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.
12	The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
13	The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.
14	The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
15	The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.
16	The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

No.	Limitation
17	The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.
18	The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.
19	The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.
20	The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

EXHIBIT 3C

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ARISTA NETWORKS, INC.,
Petitioner

v.

CORRIGENT CORPORATION,
Patent Owner.

Case No. IPR2023-00805

U.S. Patent No. 7,593,400

Petition for *Inter Partes* Review of U.S. Patent No. 7,593,400

Mail Stop **PATENT BOARD**
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450 Alexandria, VA 22313-1450

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES

OTHER AUTHORITIES

PETITIONER'S EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 7,593,400 ("the '400 patent")
1002	Copy of Prosecution History of the '400 patent
1003	Declaration of Dr. Tal Lavian
1004	U.S. Patent Application Publication No. 2005/0198371 ("Smith")
1005	Certified English Translation of Japanese Patent Application No. 2005086668 ("Ishimori")
1006	U.S. Patent No. 6,735,198 ("Edsall")
1007	U.S. Patent Application Publication No. 2004/0133619 ("Zelig")
1008	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 1998 Edition ("802.1Q")
1009	Declaration of Dr. Mary K. Bolin
1010	*Corrigent Corp. v. Dell Technologies et al.*, No. 1:22-cv-00496 (D. Del.), Dkt. 1
1011	*Corrigent Corp. v. Arista Networks, Inc.*, No. 1:22-cv-00497 (D. Del.), Dkt. 1
1012	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 1
1013	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 42
1014	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 46
1015	IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition
1016	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition
1017	IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition
1018	Kompella et al., *Virtual Private LAN Service*, IETF (December 2005)
1019	Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005)
1020	Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005)
1021	U.S. Patent No. 6,917,986 ("Mor")

1022	U.S. Patent No. 7,974,223 ("Zelig")
1023	Western District of Texas Statistics (Docket Navigator)
1024	District of Delaware Statistics (Docket Navigator)
1025	Japanese Patent Application No. 2005086668 ("Ishimori")
1026	U.S. Patent No. 6,999,418 ("Sharma")
1027	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 69
1028	*Dell Technologies Inc. et al v. Corrigent Corp.*, IPR2023-00370, Paper 1 (Dec. 23, 2022)
1029	Declaration of Nicholas Bambos, Ph.D. under 37 C.F.R. § 1.68

I. INTRODUCTION

Arista Networks, Inc. ("Petitioner") hereby petition for *inter partes* review of U.S. Patent No. 7,593,400 and seek cancellation of claims 1-20 ("the Challenged Claims") as unpatentable under 35 U.S.C. § 103.

II. MANDATORY NOTICES

A. Real Parties-in-Interest

Petitioner certifies that the real party-in-interest is Arista Networks, Inc. No unnamed entity is funding, controlling, or directing this Petition, or otherwise has had an opportunity to control or direct this Petition or Petitioner's participation in any resulting IPR.

B. Related Matters

The '400 patent is the subject of this IPR Petition is also the subject of patent litigation suits brought by Patent Owner Corrigent Corporation, including against Petitioner:

- *Corrigent Corp. v. Dell Technologies et al.*, No. 1:22-cv-00496 (D. Del.) (EX1010);
- *Corrigent Corp v. Arista Networks, Inc.*, No. 1:22-cv-00497 (D. Del.) (EX1011); and
- *Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.) (EX1012).

C. Lead and Backup Counsel

Pursuant to 37 C.F.R. §§ 42.8(b)(3), 42.8(b)(4), and 42.10(a), Petitioner provide the following designation of counsel:

Lead Counsel	Back-up Counsel
Eliot D. Williams (Reg. No. 50,822) Baker Botts L.L.P. 700 K Street, N.W. Washington, D.C., 20001-5692 Tel: 202-639-1334 Eliot.Williams@BakerBotts.com	Kurt Pankratz (Reg. No. 46,977) Baker Botts L.L.P. 2001 Ross Avenue, Suite 900 Dallas, TX 75201-2980 Tel: 214-953-6500 Kurt.Pankratz@BakerBotts.com Jeremy J. Taylor (Reg. No. 73,912) Baker Botts L.L.P. 101 California Street, Suite 3200 San Francisco, CA 94111 Tel: 415-291-6200 Jeremy.Taylor@BakerBotts.com Sean Y. Lee (Reg. No. 77,322) Baker Botts L.L.P. 401 S 1st St Suite 1300 Austin, TX 78704 Tel: 512-322-2500 Sean.Lee@BakerBotts.com

D. Service Information

Service via hand delivery or postal mail may be made at the addresses of the lead and back-up counsel above. Petitioner hereby consents to electronic service, and service via electronic mail may be made at the email addresses provided above for the lead and back-up counsel.

III. PAYMENT OF FEES

Pursuant to 37 C.F.R. §§ 42.103 and 42.15(a), the required fee is being submitted herewith.

IV. REQUIREMENTS OF *INTER PARTES* REVIEW

A. Standing

Petitioner certifies that the ’400 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims. 37 C.F.R. § 42.104(a).

B. Identification of Challenge and Relief Requested

Pursuant to 37 C.F.R. § 42.104(b), Petitioner requests the Board institute IPR of claims 1-20 of the ’400 patent under pre-AIA 35 U.S.C. § 103 on the prior art references and grounds described below:

Ground	Claims	Basis for Rejection
1	1, 3, 6, 11, 13, 16	35 U.S.C. § 103 over Smith and Sharma
2	1-3, 11-13	35 U.S.C. § 103 over Smith, Sharma, and Ishimori
3	1, 4-7, 10-11, 14-17, 20	35 U.S.C. § 103 over Smith, Sharma, Ishimori, and Edsall
4	8-9, 18-19	35 U.S.C. § 103 over Smith, Sharma, Ishimori, and Zelig
5	9, 19	35 U.S.C. § 103 over Smith, Sharma, Ishimori, Zelig, and 802.1Q

U.S. Patent Application Publication No. 2005/0198371 ("**Smith**", EX1004) qualifies as prior art under 35 U.S.C. § 102(a) and/or (e). Smith was filed on February 19, 2004 and published on September 8, 2005.

U.S. Patent No. 6,999,418 ("**Sharma**", EX1026) qualifies as prior art under 35 U.S.C. § 102(a), (b), and/or (e). Sharma was filed on December 21, 2001 and published on June 26, 2003.

Japanese Patent Application No. 2005086668 ("**Ishimori**", EX1005, EX1025) qualifies as prior art under 35 U.S.C. § 102(a) and/or (b). Ishimori was filed on September 10, 2003 and published on March 31, 2005.

U.S. Patent No. 6,735,198 ("**Edsall**", EX1006) qualifies as prior art under 35 U.S.C. § 102(a), (b), and/or (e). Edsall was filed on December 21, 1999 and issued on May 11, 2004.

U.S. Patent Application Publication No. 2004/0133619 ("**Zelig**", EX1007) qualifies as prior art under 35 U.S.C. § 102(a), (b), and/or (e). Zelig was filed on January 7, 2003 and published on July 8, 2004.

The **IEEE 802.1Q-1998 Standard** ("**802.1Q**", EX1008) qualifies as prior art under 35 U.S.C. § 102(b). Specifically, 802.1Q was published on March 8, 1999 and was publicly available no later than December 20, 2001. EX1009 ¶¶17-28.

Smith, Sharma, Ishimori, Zelig, and 802.1Q were not cited during the prosecution of the ’400 patent. Edsall was cited during prosecution; however,

Petitioner relies on Edsall only for disclosures of claim limitations that were found to be present in Edsall during prosecution—findings that the applicant did not challenge.

C. How the Challenged Claims Are to Be Construed Under 37 C.F.R. § 42.104(b)(3)

Petitioner notes that a claim construction order has been issued in a related case, where the term "said FDB" recited in Claims 1 and 11 was construed as having "plain and ordinary meaning, wherein 'said FDB' means 'said FDB of said first line card.'" EX1027, 3. The term "virtual media access control (MAC) bridge" recited in Claims 8, 11, and 18 was similarly construed as having plain and ordinary meaning. *Id.* Petitioner submits that, for the purposes of this proceeding and the grounds presented herein, no claim term requires express construction.

D. How the Challenged Claims Are Unpatentable Under 37 C.F.R. § 42.104(b)(4)

The following sections explain how the Challenged Claims are unpatentable under the statutory grounds identified above, including where each element of the claim is found in the prior art patents or printed publications.

E. Supporting Evidence Under 37 C.F.R. § 42.104(b)(5)

The exhibit numbers of the supporting evidence relied upon and the relevance of the evidence to the Challenged Claims, including an identification of specific portions of the evidence that support the challenge, are provided below. The

technical information and grounds for unpatentability are further supported by the Declaration of Dr. Tal Lavian (EX1003).

V. THRESHOLD REQUIREMENT FOR *INTER PARTES* REVIEW

Under 35 U.S.C. § 314(a), institution of *inter partes* review requires "a reasonable likelihood that the Petitioner would prevail with respect to at least one of the claims challenged in the petition." This petition meets this threshold for each ground of unpatentability.

VI. PERSON OF ORDINARY SKILL IN THE ART

The alleged invention relates to communication networks, specifically learning MAC addresses in a distributed or virtual bridge. EX1001, Abstract. A person of ordinary skill in the art ("POSITA") at the time of the alleged invention (May 2006) would have had a degree in electrical engineering or a similar discipline, with at least two years of relevant industry or research experience (or additional education). EX1003 ¶25. The relevant experience would include a working understanding of networking systems, including distributed bridges, MAC forwarding tables, LAG groups, and virtual LAN services. *Id.* Lack of professional experience can be remedied by additional education, and vice versa. *Id.*

VII. BACKGROUND IN THE RELEVANT ART

All concepts in this section were well known and widely used by POSITAs at least as of May 19, 2006. EX1003 ¶¶44-61.

A. Computer Networks

Computer networks are comprised of communications links that connect to ports of communication nodes. *Id.* ¶45. Data packets or "frames" transmit information between the nodes based on MAC (Media Access Control) addresses, which are unique identifiers assigned to network hardware. *Id.* ¶¶45,52-53. An L2 switching node maintains a MAC address table, which maps the output port for a particular frame based on its destination MAC address. *Id.* ¶¶55-56.

B. Switches and Learning MAC Tables

MAC addresses are used by L2 switches (also called bridges) to switch packets at the network node from an input port to an output port. *Id.* ¶¶52-54. A MAC bridge is a network device connects two physically separated LANs into a single logical LAN. *Id.* ¶52. To forward data packets, called "frames," a MAC bridge maintains a forwarding database (FDB) of MAC addresses and corresponding bridge ports of packets that have been received at the bridge. *Id.* ¶¶53-55. When a bridge receives a packet, it checks the destination MAC address of the packet. *Id.* ¶55. The bridge checks a MAC address table (or forwarding database) to determine whether it knows where to forward the message. *Id.* ¶55. If the destination MAC address of the packet is in the FDB, the bridge forwards the packet to the port associated with that destination MAC address. *Id.* If not, the bridge may send the message over every port to ensure it is received at the intended destination. *Id.* This

is called "flooding." *Id.* When packets are received, the bridge also "learns" information about the *source* MAC address of each packet and, if the source MAC address is not in the FDB, updates the database to reflect the association between the incoming port and the source MAC address. *Id.*

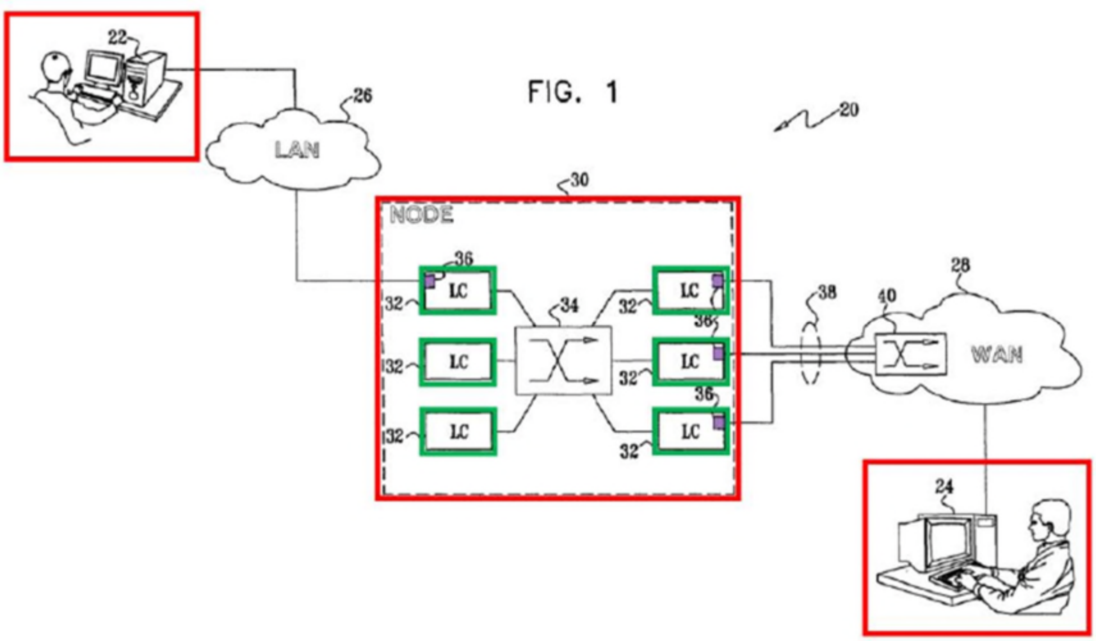
C. Link Aggregation

In the 1990s, link aggregation was used to increase bandwidth, traffic load balancing, scalability, and reliability by grouping multiple links from a switch to form a link aggregate group, or "LAG." *Id.* ¶59. These multiple links in a LAG are treated as a single logical link. *Id*. The Layer 2 Link Aggregation Sublayer combines a number of individual physical links into a single logical link, which presents a single MAC interface to the MAC client. *Id.* (citing EX1017, Section 3, Fig. 43-1).

VIII. SUMMARY OF THE ALLEGED INVENTION OF THE '400 PATENT

The '400 patent relates to communication networks, specifically learning MAC addresses in a distributed bridge. The patent acknowledges it was already well known that computing systems connect to LANs at OSI Layer 2. EX1001 1:13-19. "[A] number of authors have described methods for creating a virtual private LAN service (VPLS), which links different LANs together over an IP network." *Id.* 2:9-17. The patent explains a single provider can participate in multiple VPLSs. *Id.* 2:18-30. Treating both virtual and physical interfaces identically makes the provider

network appear as a single LAN domain. *Id.* 2:31-33. This VLAN functionality, a known technique, allows end points to be joined using link aggregation (LAG) as if they were part of the same local network even if they are not physically close to each other. *Id.* 2:37-47 (citing EX1017). For example, as shown in Fig. 1, although users of **terminals 22 and 24** are not on the same LAN domain (terminal 22 is connected to a LAN 26 and terminal 24 is connected to a wide area network (WAN) 28), the VPLS "permits the users" to "communicate with one another as though they were connected to the same LAN domain." *Id.* 5:51-62; Fig. 1.



A **network node 30** "comprises multiple **line cards 32,** linked by a switching core 34," and each line card has **ports 36,** which connect to other nodes in LAN 26 and WAN 28. *Id.* 6:8-12. The line cards may serve as both ingress and egress with a MAC forwarding database (FDB), which stores learned MAC addresses and which ports they were learned on. *Id.* 3:7-11. When an ingress line card receives an

incoming packet, it consults the FDB for the MAC destination address of the packet to determine which line card and port to forward the packet to. *Id.* 3:11-17. When the MAC destination address does not appear in the FDB, the line card floods the packet to all of the ports, as in the prior art. *Id.* The patent explains that if the packet is transmitted using a LAG, a single port within the LAG is chosen and the packet transmitted only through that port, including when the packet is flooded. *Id.* 6:31-48,7:47-49. According to the patent, however, given that the other members of the LAG may not receive such packets, their FDBs may not be updated, which may result in unnecessary flooding when dealing with future packets. *Id.* 3:34-53.

To address this, the purported invention provided "improved methods for MAC learning" that are "useful especially in the context of nodes that are configured to serve as virtual bridges in Layer 2 virtual private networks … particularly when multiple ports of the node are conjoined in a LAG group." *Id.* 2:60-3:2. The patent discloses a known technique of MAC source learning on the "egress path." *Id.* 3:18-24,FIG.3,7:55-63. The patent explains that "[l]earning on egress is advantageous particularly with respect to flooded packets, since in this case multiple line cards receive the packet and are able to learn the interface association of the MAC source address (SA) and VPLS instance." *Id.* 7:55-63.

After a line card learns the MAC source address and VPLS instance on the egress path, the line card provides a synchronization message to other line cards to

update their FDBs with the learned information. *Id.* 8:17-29,9:47-65. These synchronization messages ("SYNC") are sent at regular intervals "to report each SELF entry [*e.g.*, entries that are learned by the packet processor on the line card itself] that it has created in the FDB 58 to the other line cards 32 in node 30." *Id.* 8:17-22. The patent thus discloses a record is removed from the database "if a predetermined aging time elapses following the timestamp without a further packet having been received with the same key" and "free[s] up space for new records." *Id.* 9:4-11. If, on the other hand, the current packet matches a record in the FDB, "the packet processor refreshes the timestamp of the record," and "forwards the packet to the appropriate output port." *Id.* 9:29-31.

A. Summary of the Prosecution History of the '400 Patent

The '400 patent's application was filed on May 19, 2006. EX1001 [22]. The Examiner rejected all pending claims as anticipated by Edsall, or obvious over Edsall and U.S. Patent No. 6,788,681. EX1002 81-85. During an Examiner interview, it was discussed if the limitation "link aggregation group having a plurality of ports" in claim 1 were amended to recite the claim term as defined by the specification at 2:37-40, then the amended claim would overcome Edsall. *Id.* 112. The applicant therefore amended the claim limitation to add "a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports

and a plurality of conjoined member line cards." *Id.* 114. The applicant made no other amendments to overcome the rejections over Edsall. The Examiner allowed the claims, and the patent issued on September 22, 2009. *Id.* 128,160.

IX. GROUNDS OF UNPATENTABILITY

A. Ground 1: Claims 1, 3, 6, 11, 13, and 16 would have been obvious over the Smith-Sharma combination.

Smith in view of Sharma renders obvious claims 1, 3, 6, 11, 13, and 16.

1. Smith

Smith is directed to a virtual network device comprising interface bundles, which are managed as a single logical interface. The virtual network devices contain multiple "sub-units," which "collectively operate as a single logical network device." EX1004 Abstract,¶34. To achieve this, the "system includes a virtual link bundle" with "several communications links," and the "communications links are configured to be managed as a single link." *Id.* ¶9.

As depicted below, **virtual network device 202** is coupled to other network devices 120(1)-120(3). *Id.* ¶44. The virtual network device consists of virtual network device sub-units 122(1) and 122(2), which includes several **line cards 304(1)-304(4)**. *Id.* ¶46. The line cards include a "forwarding engine" and "**interfaces**." *Id.* ¶¶46-47. When a packet is received at an uplink interface, the virtual network device sub-unit can learn "the sending device's [MAC] address" by "associating the MAC address with the logical identifier of the uplink interface." *Id.*

¶54. The sub-unit then informs each forwarding engine of this association. *Id.* "[P]ackets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier." *Id.*

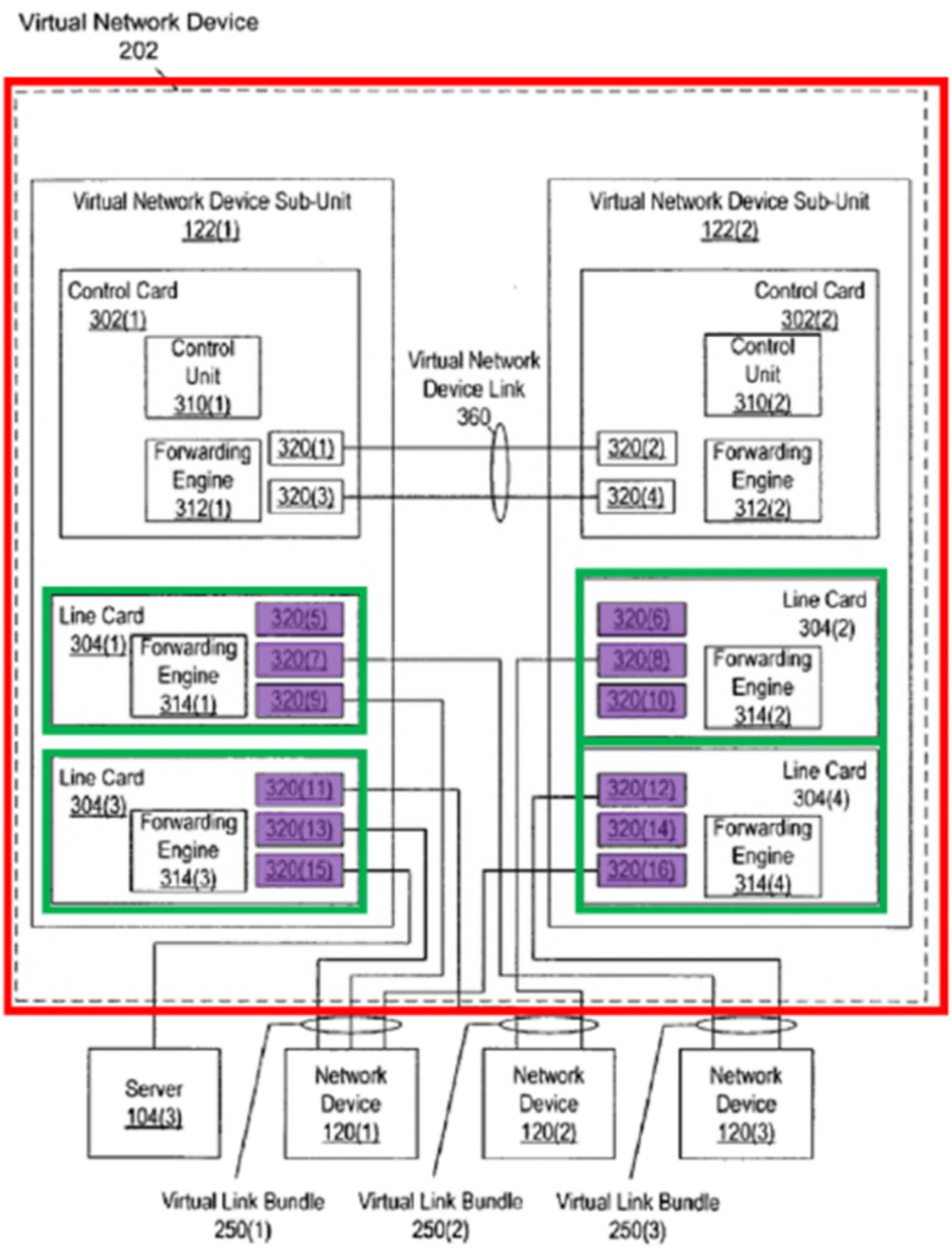


FIG. 3

Smith uses "MAC notification frames" in order to "keep the content of the L2 tables in the virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶62. If a "forwarding table already includes an entry associating the destination address with a port of one of the network devices," that forwarding engine will generate "a MAC notification identifying this association" to "any other forwarding engines within" the virtual network device sub-unit. *Id.* ¶63. "If there is no hit in the forwarding table," "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN" excluding the "interface that the packet arrived on." *Id.* ¶66. As a result of the virtual link bundling, the data packet is sent via only "one of the communication links." *Id.* ¶9.

2. Sharma

Sharma is directed to a technique for "reduc[ing] the incidence of unnecessary frame flooding," particularly "in connection with aggregated ports." EX1026 2:11-18. Sharma's framework utilizes a known learning process referred to as "egress learning," where a frame's source address is learned as the frame exits a network device through an egress port, as opposed to learning at an ingress port. *Id.* 1:60-2:7,1:41-51,4:6-11,3:3-4.

An aggregated port ("AP") is a set of physical ports that are logically grouped together such that the grouped ports appear as a single logical connection. *Id.* 1:52-

59. According to Sharma, when employing an AP, a particular port may be selected for transmitting frames in one direction while a different port may be selected for transmitting frames in the other direction. *Id.* 4:33-5:10. In such a scenario, address information learned by one port of the AP may not be available to another, and thus frames sent through the AP may be continuously flooded "in a wasteful manner, despite the existence of information at another port of the device that could be used to terminate the flooding." *Id.* 1:60-2:7. One known solution to this issue is to periodically synchronize the forwarding tables of all ports, such that the information learned by one port is shared with other ports. *Id.* 5:11-21. However, Sharma states that "such a mechanism is resource-intensive," and "[r]eliance upon this mechanism alone may result in relatively inefficient operation." *Id.* 5:21-25. The purported invention of Sharma provides another solution, where the forwarding tables are selectively synchronized by counting the number of instances a particular port floods a frame with an unknown destination address, and if that number reaches a predetermined threshold, inferring that another port has the necessary information about that destination address, and sharing that information with the particular port. *Id.* 2:19-42.

3. Motivation to combine Smith and Sharma

A POSITA considering the teachings of Smith would have also considered the teachings of Sharma, as they are analogous prior art pertaining to the same field

of endeavor, namely, forwarding data packets through a communication network employing aggregated, or bundled, ports and links. *Compare* EX1004 ¶¶2,6 *with* EX1026 1:16-19,1:52-59. A POSITA would have been motivated to combine the teachings of Smith and Sharma for multiple reasons, as discussed below. EX1003 ¶98.

First, the framework of Sharma's invention is substantially identical to that of Smith. Smith discloses techniques for forwarding a data packet through a network device employing aggregated links. EX1004 ¶¶6,9,30,50. Sharma similarly discloses techniques for forwarding a data packet through a network device employing aggregated ports. EX1026 1:20-28,1:52-59,4:33-5:25. The only difference between the two disclosures is the use of alternatives source learning processes. Sharma's framework utilizes a known learning process referred to as "egress learning," where a frame's source address is learned as the frame exits a network device through an egress port. *Id.* 1:60-2:7,1:41-51,4:6-11,3:3-4. In contrast, Smith's framework learns a frame's source address when the frame is first received by a network device (referred herein as "ingress learning"). EX1004 ¶0065. Egress learning was a known substitute of ingress learning. EX1003 ¶98. There is nothing uniquely challenging or difficult about modifying a network device to learn a frame's source address when the frame is first received by the network device as opposed to when the frame is to be transmitted out of the network device. EX1003

¶98. A POSITA would have been motivated to combine the teachings of Smith and Sharma based on simple substitution and would have had a reasonable expectation of success doing so. EX1003 ¶¶98-100.

Second, it would also have been obvious to combine the teachings of Sharma and Smith because Sharma expressly identifies a technique used by Smith and provides improvements to it. Sharma explains that the traditional methods for synchronizing the forwarding tables are inefficient and wasteful, particularly because flooding itself is resource intensive. EX1026 5:16-33. Smith's framework utilizes these traditional methods. EX1004 ¶63. Sharma then provides an improved way of synchronizing the forwarding tables by selectively updating them by counting the number of instances a particular port floods a frame with an unknown destination address, and if that number reaches a predetermined threshold, inferring that another port has the necessary information about that unknown destination address, and instructing that port to share that information. EX1026 2:19-42. A POSITA looking to improve the efficiency of operating a network device would have recognized the problem that Sharma described and further recognized that Smith's system also shared that problem and was ready to be improved using Sharma's technique of selectively updating the forwarding tables in Smith's system. EX1003 ¶100.

4. Analysis of Ground 1

a) Claim 1[pre]: A method for communication, comprising:

To the extent the preamble is limiting, Smith discloses claim 1[pre]. Smith's "system includes a virtual link bundle, which includes several communication links," which constitute methods for communication. EX1004 ¶9; EX1003 ¶114.

b) Claim 1[a]: configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;

Smith discloses a **virtual network device 202**, which is a network node. EX1004 ¶36. The virtual network device "includes **several cards**" such as 304(1) and 304(4) (e.g., first and second line cards). *Id.* ¶46. These line cards include **interfaces**, for example, 320(5), 320(7), and 320(9) on line card 304(1), and 320(12), 320(14), and 320(16) on line card 304(4) (e.g., a plurality of ports). *Id.* ¶47. Thus, Smith's first and second line cards have respective first and second ports. EX1003 ¶115.

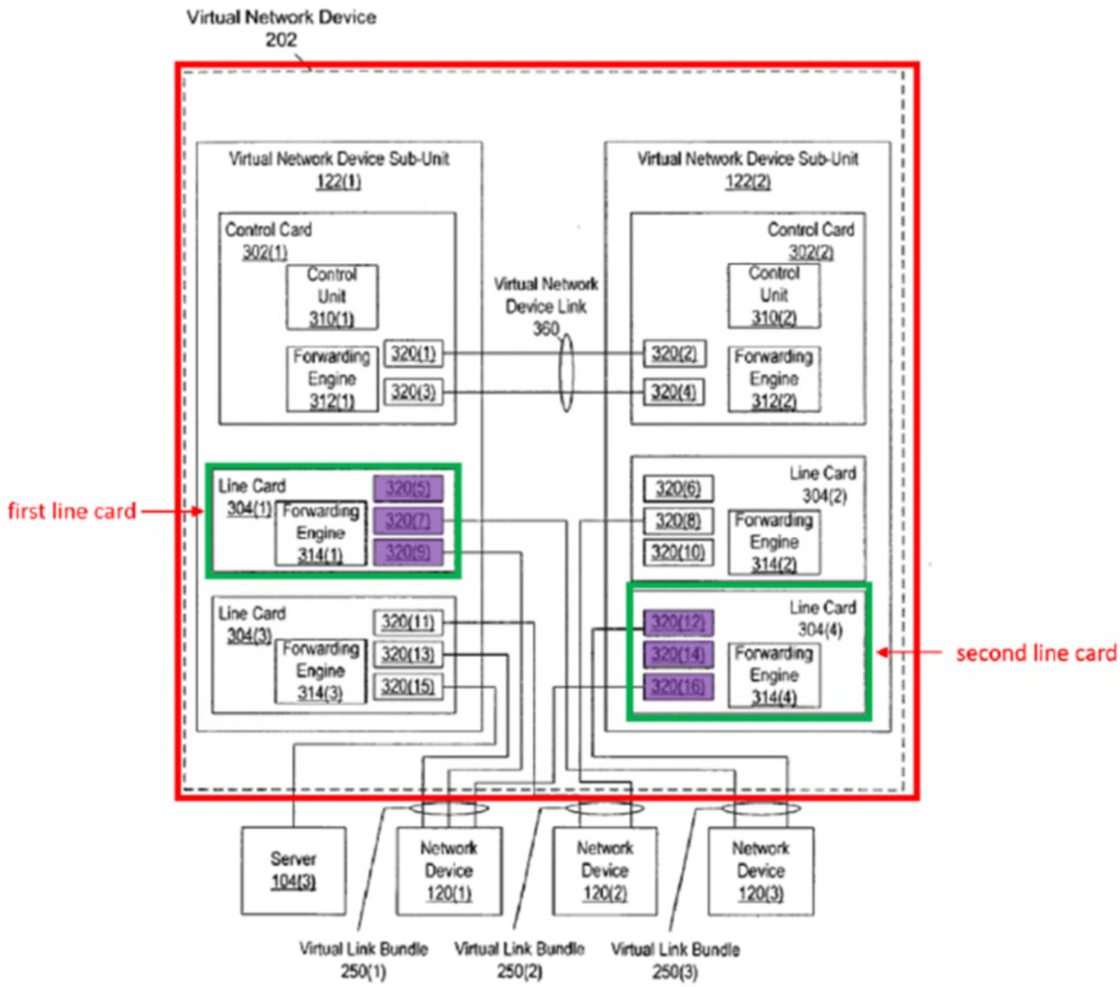


FIG. 3

An "interface" and "port" are synonymous and Smith treats them interchangeably, and thus, discloses the first and second ports. EX1004 ¶63 (stating "port *or* uplink interface"). Smith teaches that the virtual link bundles, which are a link aggregation (LAG) group of parallel physical links, may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶6,9,30. The virtual network device 202 "route[s] and forward[s] packets to and from network devices 120(1)-120(3)" by associating the MAC address of a received data packet with the logical

identifier of the uplink interface. *Id.* ¶54. Smith's virtual network device 202 therefore operates as a distributed MAC bridge in a Layer 2 data network. EX1003 ¶¶116-118.

c) Claim 1[b]: configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards;

As stated above, Smith's virtual link bundles, which are a LAG group of parallel physical links, may provide Layer 2 forwarding, and are "managed as a single link." EX1004 ¶¶6,9,30. Smith discloses a **network device 120(2)** (e.g., one endpoint) "coupled to **virtual network device 202**" (e.g., another endpoint) "by **virtual link bundle 250(2)**" (e.g., a LAG group) as shown in the annotated figure below. *Id.* ¶44. The virtual link bundle 250(2) consists of **two uplinks** connected to respective **ports** (e.g., a plurality of LAG ports). *Id.* ¶51.

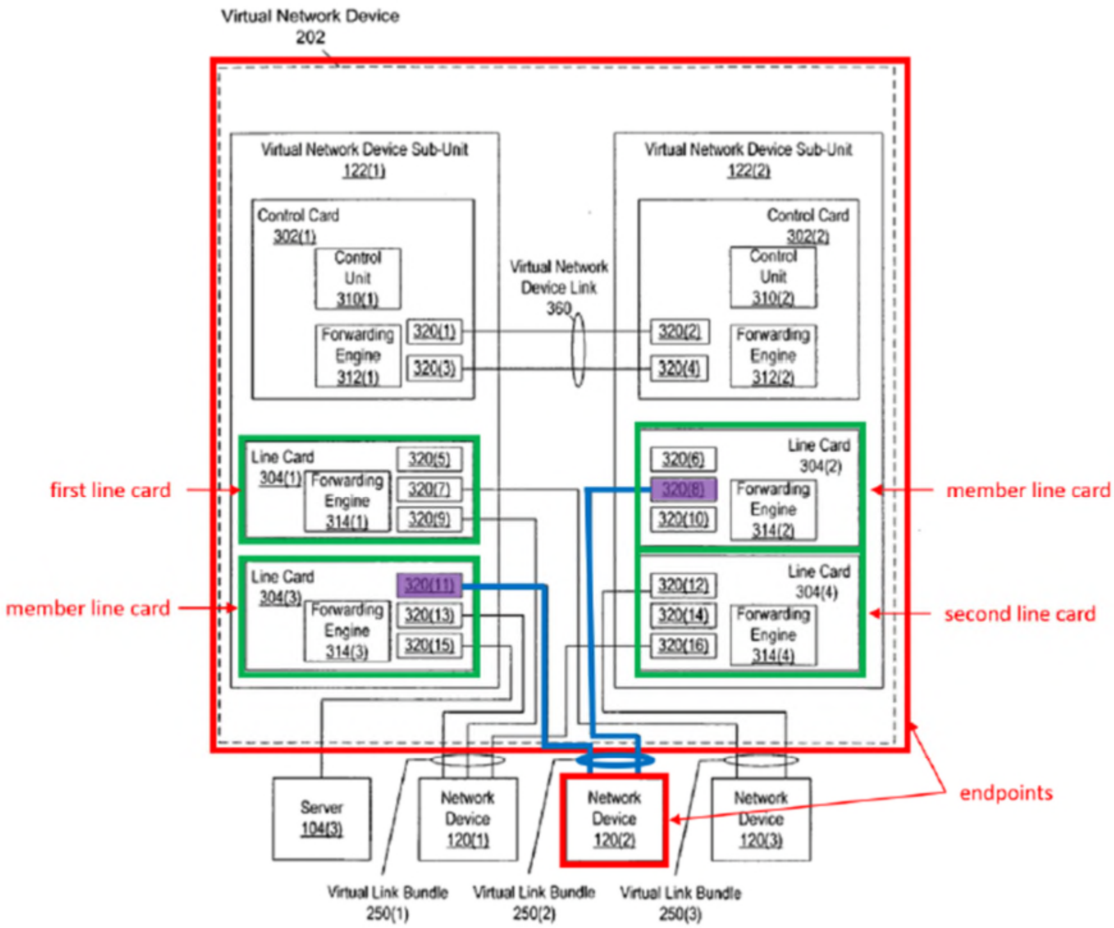


FIG. 3

Smith further specifies the virtual network devices provide Layer 2 "forwarding and routing." *Id.* ¶30. Because virtual network device sub-units 122(1) and 122(2) "can coordinate their behavior such that they appear to be a single virtual device," the remaining line cards 304(2) and 304(3) comprise the conjoined member line cards. *Id.* ¶50. Smith therefore discloses configuring a link aggregation (LAG) group of parallel physical links (e.g., **virtual link bundle 250(2)**) between two endpoints (e.g., **virtual network device 202** and **network device 120(2)**) in said Layer 2 data network joined together into a single logical link, said LAG group

having a plurality of LAG ports (e.g., **interfaces 320(8) and 320(11)**) and a plurality of conjoined member line cards (e.g., **line cards 304(2) and 304(3)**). EX1003 ¶¶119-121.

d) Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

The member line cards in Smith include a **forwarding engine** and **interfaces** (e.g., ports). EX1004 ¶47.

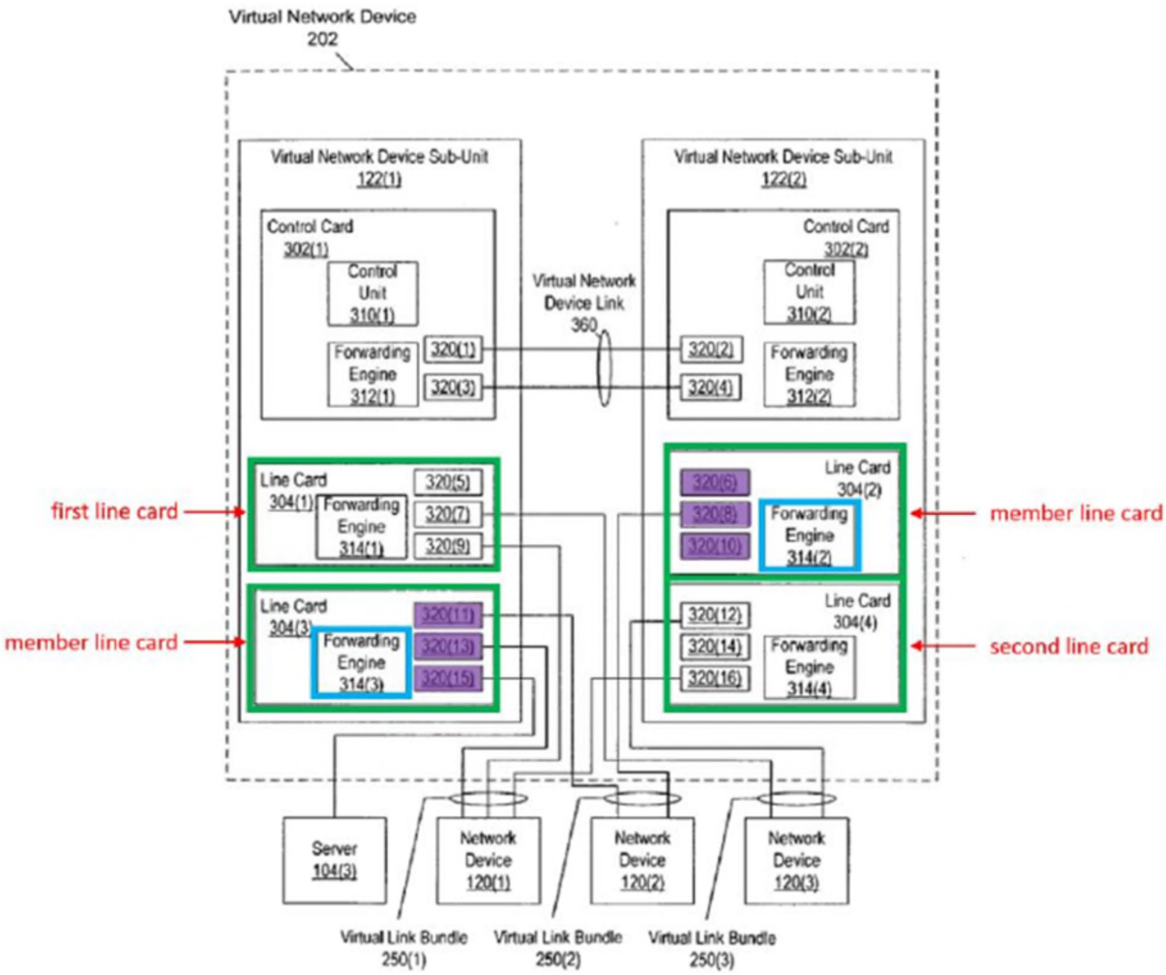


FIG. 3

When a packet is received on a particular uplink interface, the virtual network device learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] **uplink interface**" (e.g., associating MAC addresses with ports of said plurality of ports of said network node). *Id.* ¶54. For subsequent packets that are addressed to one of the learned MAC addresses, the "**forwarding engine**" uses the learned association to identify the appropriate logical identifier and routes/forwards the packet accordingly. *Id.* Smith teaches this

information may be used to "set up or modify lookup tables" (e.g., the forwarding database (FDB)). *Id.* ¶¶57,61. Moreover, FIG. 2 illustrates each of the **line cards** having a respective **forwarding engine**. Thus, Smith teaches the lookup table (e.g., FDB) may store records associating MAC addresses with the logical identifier of the uplink interface (e.g., ports of said plurality of ports) of said network node. EX1003 ¶¶122-123.

e) Claim 1[d]: receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;

Smith discloses that the uplink interface (e.g., ingress port) receives a data packet with the "sending device's MAC address" (e.g., a MAC source address). EX1004 ¶54. Based on Figure 3, one such ingress port from a data packet received from **network device 120(1)** (e.g., MAC source address) would be **interface 320(13)**.

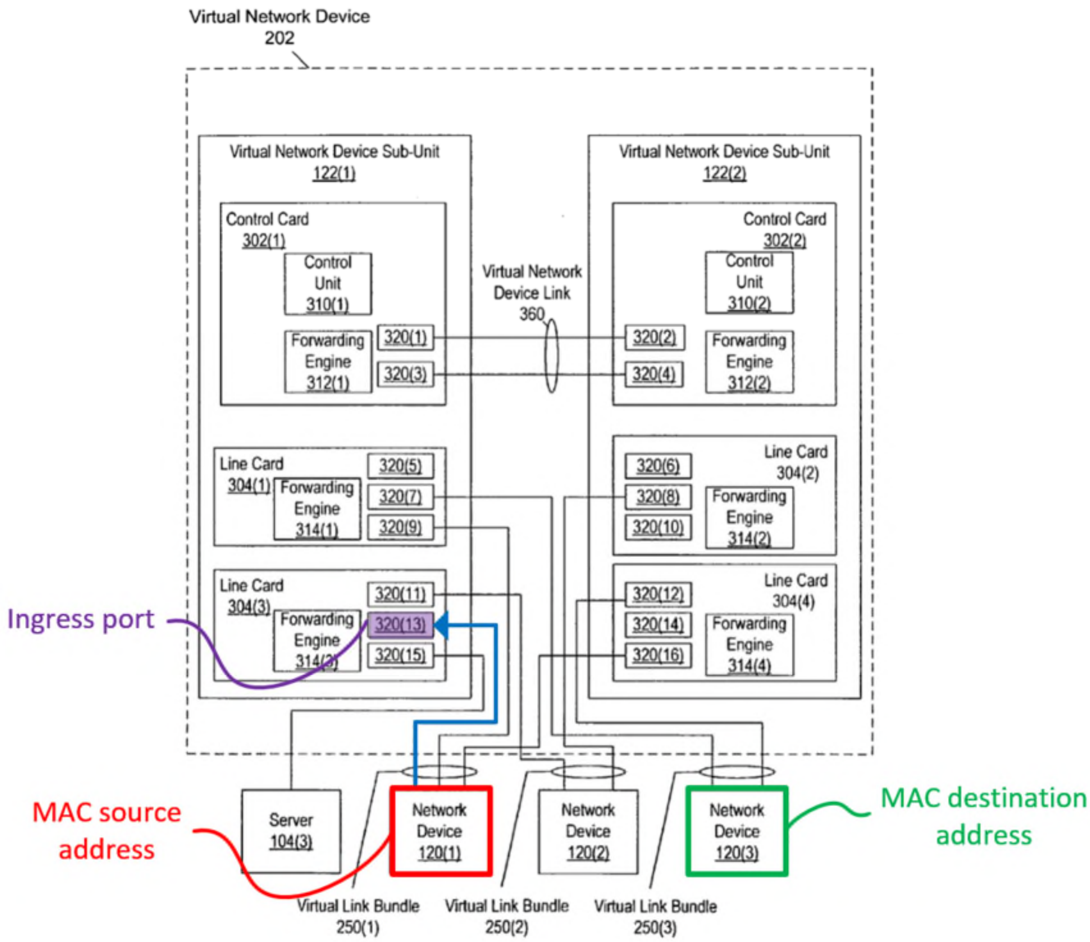


FIG. 3

Smith therefore teaches receiving a data packet on an ingress port (e.g., **interface 320(13)**) of said network node (e.g., virtual network device 202) from a MAC source address (e.g., **network device 120(1)**). Smith further discloses this data packet has a "destination logical identifier," such as the MAC destination address of **network device 120(3)**. *Id.* ¶¶60,62. Smith specifies these devices provide Layer 2 forwarding. *Id.* ¶30. Thus, Smith teaches this limitation. EX1003 ¶¶124-127.

f) Claim 1[e]: conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to

at least said first line card for transmission to said MAC destination address;

Smith teaches conveying a packet to the appropriate uplink interface (port) based on the MAC destination address. EX1004 ¶54; *supra* §IX.A.4.d); EX1003 ¶¶128-129. Referring again to Figure 3, in order to transmit the data packet to the MAC destination address (e.g., network device 120(3)), the data packet would be conveyed from interface 320(13) to **interface 320(7)** (e.g., said first port). **Interface 320(7)**, which is on **line card 304(1)** (e.g., said first line card), would then transmit the data packet to the MAC destination address (e.g., network device 120(3)).

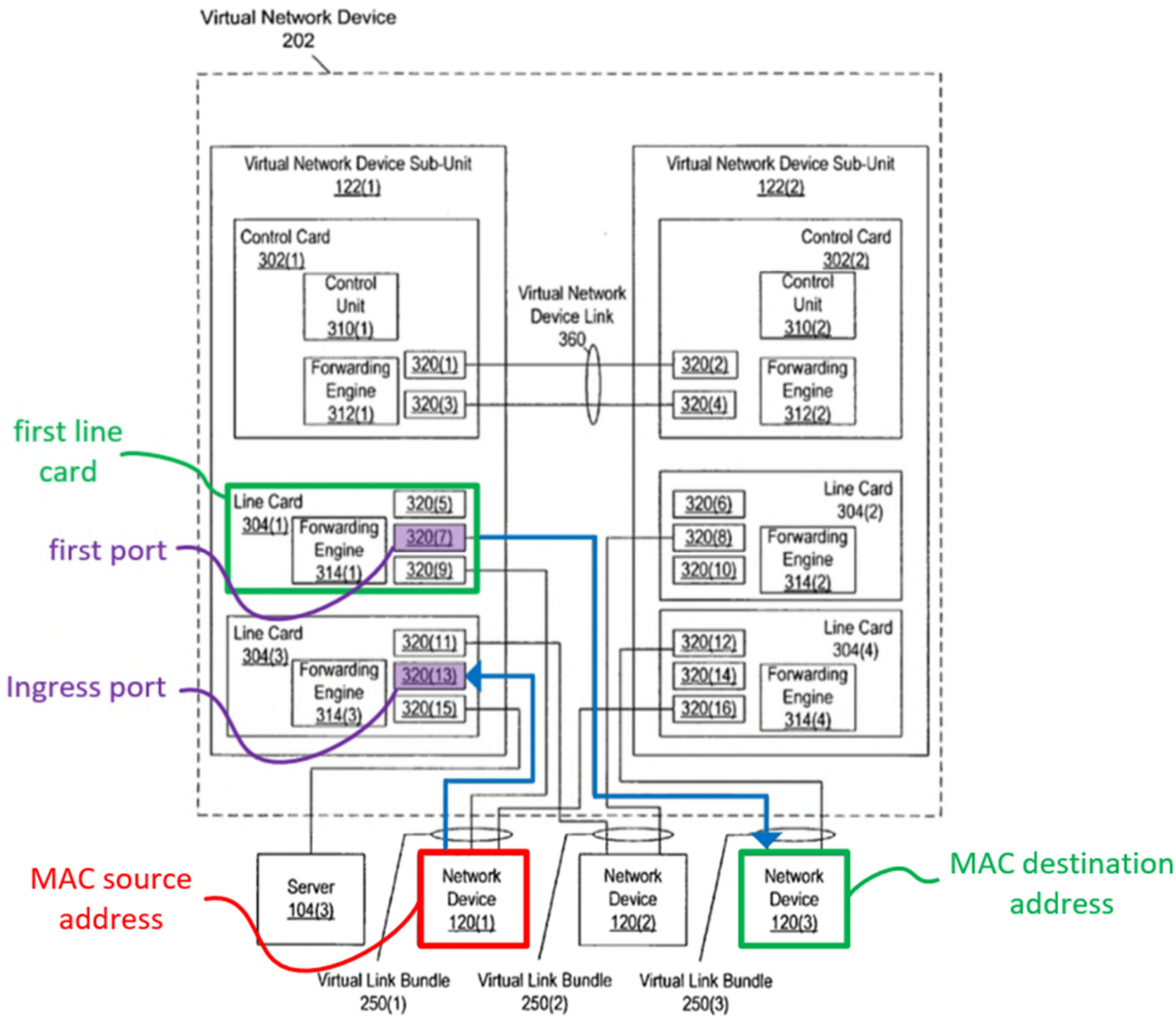


FIG. 3

Smith also teaches that particular interfaces may "act as ingress-only or egress-only." EX1004 ¶49; EX1003 ¶130. Accordingly, if a packet is received on an interface designated as "ingress-only" (e.g., **ingress port, or interface 320(13)**), the packet would then have to be conveyed to another interface that could perform egress functions (e.g., **first port, or interface 320(7)**). Moreover, Smith teaches that its redundant architecture allows the network device 202 to continue operating

even if some of its interfaces were to fail. For example, if a packet is received on a particular interface that fails and is no longer able to output the packet, the packet would have to be conveyed to another interface that is able to output the packet. *Id.*

g) Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;

Smith teaches that the virtual network device sub-unit 122(1) looks up the destination address in a "lookup table" (e.g., FDB). EX1004 ¶61; EX1003 ¶¶131-133. If there is no hit in the forwarding table, "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN." EX1004 ¶¶63,66. Smith further discloses "flooding … via one and only one LAG port." In particular, Smith teaches when the network device sends a packet "via the virtual link bundle," it "selects one of the communication links on which to send the packet." *Id.* ¶9; EX1003 ¶133; *see also* EX1026 1:57-59.

h) Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

Claim 1[g] is addressed with claim 1[h] below (§IX.A.4.i)).

i) Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the

association to each member line card of said plurality of member line cards.

Consistent with the language of claim 1[e] (§ IX.A.4.f), the "first line card" recited in claims 1[g] and 1[h] is a line card that the data packet is conveyed to "for transmission to said MAC destination address," or otherwise referred herein as an "egress line card." EX1003 ¶136.

Sharma discloses a technique referred to as "egress learning," where the source address is learned on the line card at which the packet is to be transmitted out of the network device, rather than the line card at which the packet is first received by the network device. EX1026 4:6-23. Egress learning involves "search[ing] of the egress port's forwarding table" to check whether an entry corresponding to the source address of a frame (data packet) can be found. *Id.* If an entry exists, the entry is updated with "the identity of the ingress port from which the frame was received." *Id.* If an entry is not found, "a new entry is created and added to the table," which specifies "the SA and an identifier of the ingress port from which the egress port received the frame." *Id.* 4:24-32. Sharma further discloses that each of its ports have an associated forwarding table, and that each port can "reside on different line cards." *Id.* 3:45-50,5:11-14. Thus, Sharma discloses checking the MAC source address of the data packet against the FDB of the first line card (on the egress port), and updating the FDB if the MAC source address and its association with the ingress port is not already in the FDB. EX1003 ¶137.

Sharma further discloses that entries in a forwarding table can be periodically synchronized with other forwarding tables. EX1026 5:11-21. Smith similarly teaches that when a particular forwarding engine learns of a new source address association with an ingress port, a MAC notification (e.g., a message) is sent to other forwarding engines to allow them to also learn the same. EX1004 ¶63. Thus, Smith and Sharma both disclose sending a message of the learned association to other line cards in the network device. EX1003 ¶138.

j) Claim 3: The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

Smith discloses that the network device sub-unit 122(2) sends a MAC notification (e.g., the message) to update the forwarding engines. EX1004 ¶63. "After being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* A POSITA would understand this includes the other line cards in virtual network device 202, such as line card 304(4). EX1003 ¶142. For example, line card 304(4) (e.g., the second line card) would receive the MAC notification (e.g., the message) and update its lookup tables. EX1004 ¶57. Thus, Smith discloses in response to the message, the record of

association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card. EX1003 ¶¶142-143.

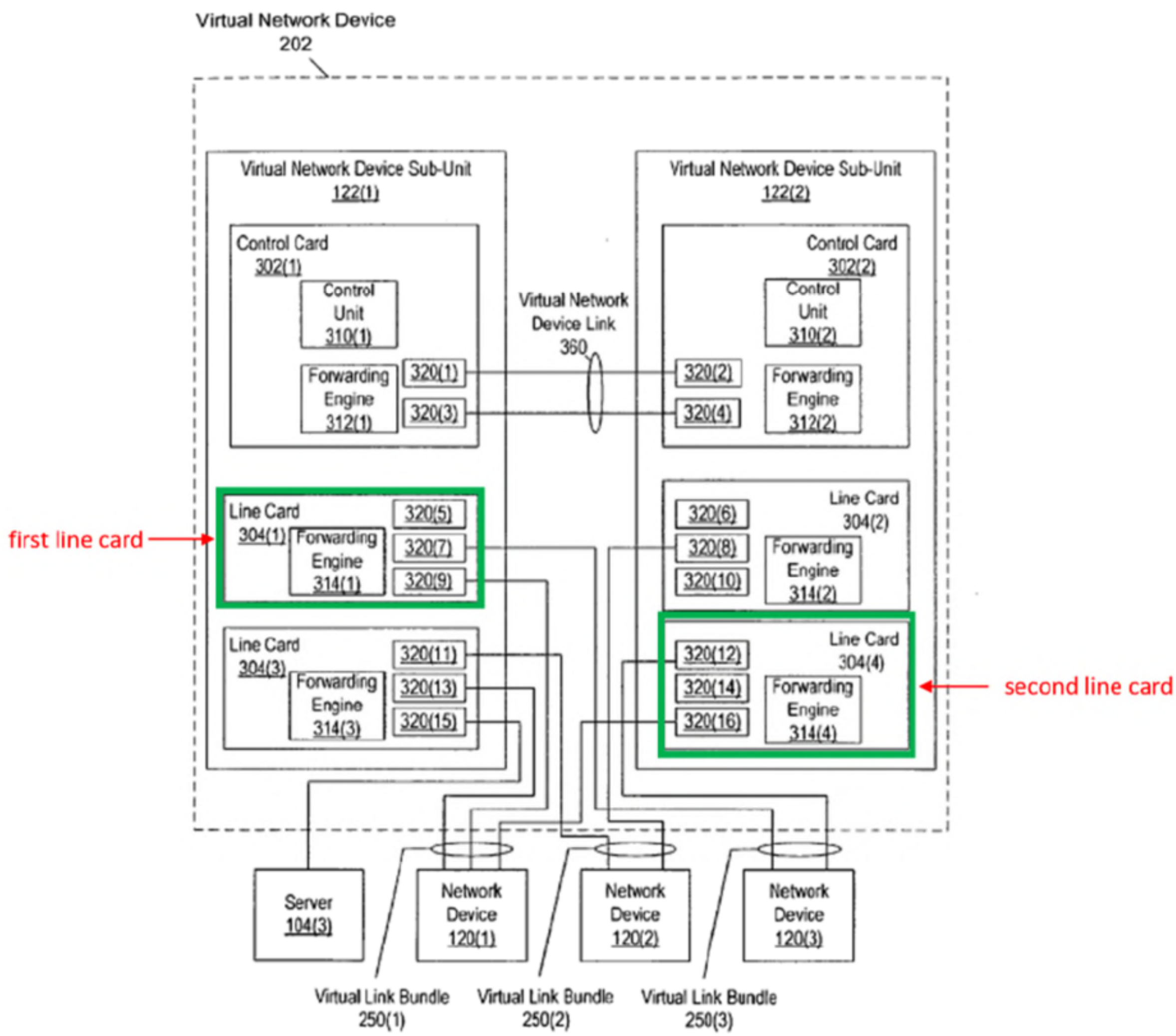


FIG. 3

Sharma discloses that entries in a forwarding table can be periodically synchronized with other forwarding tables. *Id.* 5:11-21.

k) Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via switching core of the network node to at least the second line card.

Smith discloses that the MAC notification frames are used to keep the content of the L2 tables synchronized. EX1004 ¶62. Thus, it would have been obvious to a POSITA to implement the MAC notification frame as a synchronization packet and transmit it to at least the second line card via the switching core. EX1003 ¶¶145-147.

l) Claim 11[pre] 11: A node for network communication, comprising:

To the extent the preamble is limiting, Smith discloses claim 11[pre]. Smith's system contains communications links that are coupled to network devices, which is a node. EX1004 ¶9; EX1003 ¶148.

m) Claim 11[a]: a switching core;

Smith discloses a switching mechanism that allows line cards to communicate with each other. EX1004 ¶62. A switching core was a well-known technique for line cards to communicate with one another in a switching device. EX1003 ¶ 149. It would be within the knowledge of a POSITA that the line cards in Smith would be communicating with each other via a switching core. *Id.*

n) Claim 11[b]: a plurality of ports;

The network devices in Smith contain several ports (e.g., a plurality of ports). *Id.* ¶150; EX1004 ¶6 ("EtherChannel (TM) port bundle can be formed ***from several ports on a switch***....").

o) Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of

parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

See §§IX.A.4.b)-IX.A.4.d),IX.A.4.m); EX1003 ¶151.

p) Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

See §§IX.A.4.e)-IX.A.4.i); EX1003 ¶152.

q) Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of

the second line card if the record does not already exist in the FDB of the second line card.

See §IX.A.4.j); EX1003 ¶153.

r) Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

See §IX.A.4.k); EX1003 ¶154.

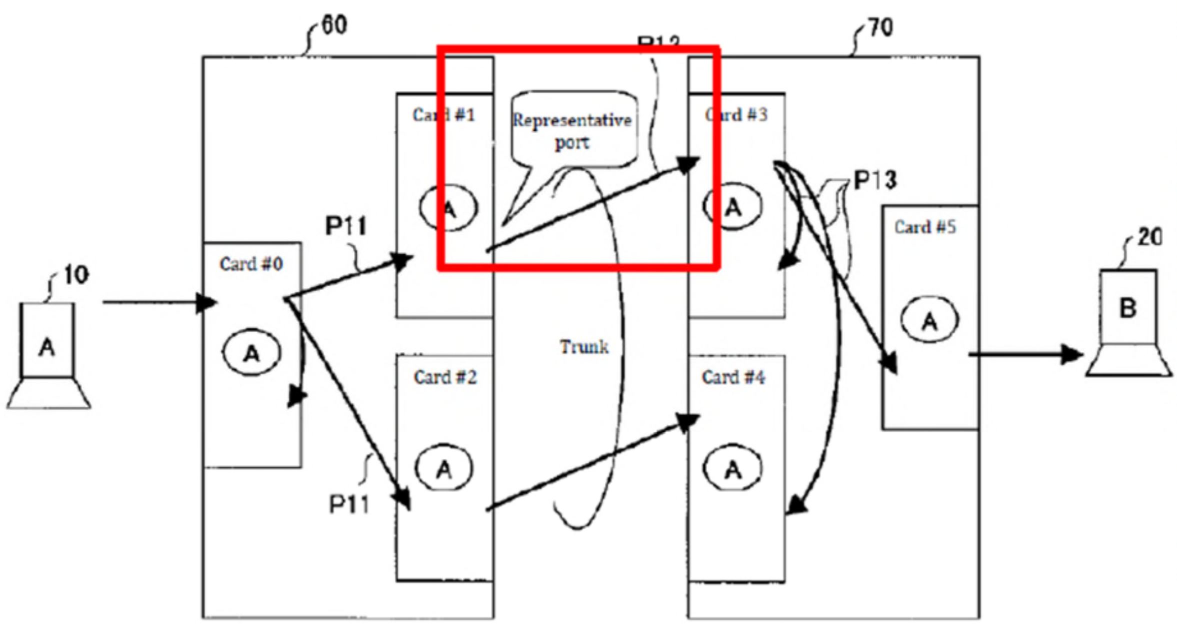
B. Ground 2: Claims 1-3 and 11-13 would have been obvious over the Smith-Sharma-Ishimori combination.

To the extent the Smith-Sharma combination does not render obvious claims 1, 3, 11, and 13, it would have been obvious in further view of Ishimori. Petitioner incorporate by reference the analysis of Smith's and Sharma's disclosures from above. §IX.A.4. The Smith-Sharm-Ishimori combination also renders obvious claims 2 and 12.

1. Ishimori

Ishimori is directed to a packet forwarding method that uses "path learning" and "link aggregation" to "prevent repeated flooding." EX1005, Abstract. Ishimori's system has node groups 60, 70 in between two terminals A10 and B20. *Id.* Fig. 9. The node groups 60, 70 have communication cards #0-2 and #3-5, respectively. *Id.* ¶15. Ishimori teaches a method where the source MAC address of a received packet is learned by storing the source MAC address "in a buffer [e.g., MAC table] had by each node." *Id.* ¶2. Thus, the destination MAC address for a

packet is searched for in the MAC table of the local device. *Id.* ¶4. If the result is found in the buffer (e.g., MAC table), then the packet is transmitted according to the forwarding path that was learned. *Id.* Thus, "as long as a learning result relating to the destination MAC address (DA)" exists in the buffer, "the forwarding path to apply to the corresponding packet can be decided unambiguously by using this information," and "there is no need for flooding." *Id.*



[FIG. 9]
Block diagram for describing one example of conventional link aggregation function

However, if the MAC address is not found, Ishimori acknowledges the inefficiency of flooding to all nodes, because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly

performed." *Id.* ¶¶2,19. Ishimori's solution to this flooding problem is the same solution the '400 patent later claimed. Specifically, Ishimori teaches, by leveraging "link aggregation," which bundles a plurality of ports to function as one virtual port, that "one representative port is selected from among the large number of ports." *Id.* ¶13. And the packet is therefore only flooded to "this **representative port**." *Id.* "[A] packet received from the terminal 10 and addressed to terminal 20 is received at card #0" of node 60. *Id.* ¶15. "One representative port is selected" and "this packet is further transmitted." *Id.* Additionally, "each communication card #0 to #5 of the node groups 60, 70" learns "information relating to the path leading to the location device" in "association with the source address." *Id.* ¶16. Ishimori teaches a learning method that "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." *Id.* ¶25. The "learn packet," which informs the plurality of line cards regarding new associations, "is transmitted to all nodes having the corresponding trunk." *Id.* ¶33.

Ishimori also teaches an aging process. When the buffers of the cards of each node learns a MAC address, the "hit bit" corresponding to the learned MAC address is set to "1." *Id.* ¶9. Then, after the first aging cycle, the hit bit is set to "0," which indicates the MAC address should be deleted from the buffer on the next cycle. *Id.* If the MAC address is learned again at the next aging process, the hit bit is reset to

"1." *Id.* This prevents "performing flooding repeatedly and thus inviting increased line traffic." *Id.* ¶25.

2. Motivation to combine Smith and Sharma with Ishimori

A POSITA would have been motivated to combine Smith, Sharma, and Ishimori, because all references disclose link bundling and path learning techniques in communication networks. As explained above, Smith discloses sending a packet to only "one of the communication links." EX1004 ¶9. Sharma discloses the same. EX1026 1:57-59. Smith further teaches for interfaces included in interface bundles, "the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet." EX1004 ¶66. The purpose of flooding over only one link in a bundle is to prevent sending packets over unnecessary links. EX1003 ¶¶101-102. For example, a POSITA would have understood flooding would not occur over the link from which the packet was received. *Id*. ¶102.

Similarly, to the extent Smith or Sharma does not explicitly disclose flooding via only one port in a LAG, Ishimori discloses another way of decreasing the number of links over which flooding occurs by explaining that flooding in a LAG need only happen over one representative port. EX1005 ¶13. Ishimori recognized the shortcomings in the approach of flooding all ports because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶19. Ishimori explains using "one representative port"

prevents the application of "a different path in the same trunk to the same packet." *Id.* ¶¶13-14. A POSITA looking to solve the shortcomings of flooding all the links would have therefore looked to Smith and Ishimori. EX1003 ¶¶101-102. Smith discloses the concept of selecting one egress interface on a virtual network bundle, and Ishimori provides using "one representative port" to prevent constant flooding.

Additionally, Ishimori teaches a "learn packet" is generated at a "predetermined timing." EX1005 ¶25. A POSITA looking to limit flooding over unnecessary links would have therefore been motivated also to implement Ishimori's learn packet generated at a predetermined timing into Smith's MAC notification method. EX1003 ¶104. It would have been within the knowledge of a POSITA that implementing such a method to make this modification and combine the teachings of Smith and Ishimori would have a reasonable expectation of success. EX1003 ¶¶101-102.

Additionally, Sharma and Ishimori are both assigned to the same assignee, and so a POSITA would have had a motivation to combine these references for this additional reason. *Abbot Vascular, Inc. v. Flexstent*, IPR No. 2019-882, Paper 48, 28-29 (Oct. 2, 2020); *Laird Techs., Inc. v. Garftech Int'l Holdings, Inc.*, IPR No. 2014-24, Paper 46, 30-31 (Mar. 25, 2015).

3. Analysis of Ground 2

a) Claim 1[c]

To the extent this limitation is not disclosed by Smith—§IX.A.4.d)—it is disclosed by the combination of Smith and Ishimori. Each of Ishimori's communication cards has a "MAC table." EX1005 ¶2. Ishimori explains when a packet from terminal A is received at port #0 of card #0, the path information for address "A" is learned by storing the information in the MAC table "had by this card" (e.g., card #0). *Id.* ¶3. Ishimori's line cards therefore maintain a MAC table (e.g., FDB) to store path information between MAC address and the receiving card and port (e.g., hold records associating MAC addresses with ports of said plurality of ports of said network node). *Id.* ¶3; EX1003 ¶¶156-158. It would have been obvious to substitute the line cards from Ishimori into Smith, and a POSITA would have had a reasonable expectation of success in making this simple substitution. *Id.* ¶155.

b) Claim 1[f]

To the extent this limitation is not disclosed by Smith or Sharma—§IX.A.4.g)—it is disclosed by the combination of Smith, Sharma, and Ishimori. Ishimori expressly discloses flooding is performed "when an address is not learned" on "one representative port" (e.g., via one and only one LAG port). EX1005 ¶13. A POSITA would have been motivated to combine Smith with Ishimori because Ishimori explains using "one representative port" prevents "the application of a

different path in the same trunk to the same packet." *Id.* ¶¶13,14; EX1003 ¶¶159-160. Thus, a POSITA would have looked to Smith, Sharma, and Ishimori, which explains why it would be beneficial to flood only one representative port, to solve the shortcomings of flooding all ports. EX1003 ¶¶103,160. Furthermore, a POSITA would have had a reasonable expectation of success in combining Smith and Sharma with Ishimori because it would have been a simple application of Ishimori's methods on Smith's, or Sharma's, virtual network device. *Id.* ¶¶88,141.

c) Claim 1[g]

To the extent this limitation is not disclosed by Smith or Sharma—§IX.A.4.h), §IX.A.4.i)—it is disclosed by the combination of Smith, Sharma, and Ishimori. Ishimori teaches the source MAC address is stored with the reception path information including the card number and port number. EX1005 ¶3. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.*; EX1003 ¶¶161-162.

d) Claim 1[h]

To the extent this limitation is not disclosed by Smith or Sharma—§IX.A.4.h),§IX.A.4.i)—it is disclosed by the combination of Smith, Sharma, and Ishimori. Ishimori discloses the source MAC address is stored with the reception

path information including the card number and port number. EX1005 ¶3. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.*; EX1003 ¶164.

Ishimori further teaches a learning process where "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." EX1005 ¶25. Ishimori teaches the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶33. Ishimori therefore discloses sending a message of the association to each member line card for said plurality of member line cards. EX1003 ¶¶163-164. It would have been obvious to a POSITA to implement the MAC notification in Smith to perform the method taught in Ishimori to first check whether the MAC address is found in the MAC table, and if not, create a new record of the association. *Id.* Both Smith and Ishimori teach a message of this association is sent to the plurality of member line cards. EX1004 ¶63; EX1005 ¶16.

e) Claim 2: The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the

second line card of new associations between the MAC address and the respective ports.

Smith teaches the network device sub-unit 122(2) sends a MAC notification (e.g., a message) to update the forwarding engines, but it does not disclose doing so periodically at predefined times. EX1004 ¶63. However, Ishimori discloses that a "learn packet" (e.g., message) is generated at a "predetermined timing" (e.g., periodically at predefined times). EX1005 ¶25, Fig. 15. Ishimori teaches the flooding operation informs the plurality of line cards regarding new associations "is transmitted to all nodes in the corresponding trunk." *Id.* ¶33. Thus, it would have been obvious to a POSITA that this must include at least the second line card. *Id.*; EX1003 ¶¶163-164. Sharma also discloses this. EX1026 5:11-21.

f) Claim 3

To the extent this limitation is not disclosed by Smith or Sharma—§IX.A.4.j)—it is disclosed by the combination of Smith, Sharma, and Ishimori. Ishimori teaches the reception path information is associated with each other and learned, which is done by storing the record of association to the MAC table (e.g., adding the record of the association to the FDB). EX1005 ¶2. The routes are transmitted via a "learn packet" that can "perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably." *Id.* ¶25. Ishimori discloses "a learn packet is transmitted to all nodes having the corresponding trunk," which necessarily includes at least the second line card. *Id.*

¶33; EX1003 ¶¶167-169. In Ishimori, each of the communication cards, which includes the second line card, would add a record of the association to the FDB in response to the learn packet. EX1005 ¶25. It would be obvious to a POSITA to use Ishimori's learn packet in place of Smith's MAC notification frames to send a message to the second line card to add an entry in its MAC table. EX1003 ¶166.

g) Claim 11[c]

See §§IX.A.4.b)-IX.A.4.d),IX.A.4.m),IX.B.3.a); EX1003 ¶170.

h) Claim 11[d]

See §§IX.A.4.e)-IX.A.4.i),IX.B.3.b)-IX.B.3.d); EX1003 ¶171.

i) Claim 12: The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.

See §IX.B.3.e); EX1003 ¶172.

j) Claim 13

See §IX.B.3.f); EX1003 ¶173.

C. Ground 3: Claims 1, 4-7, 10-11, 14-17, and 20 would have been obvious over the Smith-Sharma-Ishimori-Edsall combination.

To the extent the Smith-Sharma-Ishimori combination does not render obvious claims 1-3, 6, 11-13, and 16, they would have been obvious in further view of Edsall. Petitioner incorporates by reference the analysis of Smith, Sharma, and

Ishimori from above. §IX.B.3. The Smith-Sharma-Ishimori-Edsall combination also renders obvious claims 4-5, 7, 10, 14-15, 17, and 20.

1. Edsall

Edsall is generally directed to techniques for updating and synchronizing "forwarding tables contained on line cards that are interconnected by a switch fabric of a distributed network switch." EX1006 Abstract. The "forwarding table" has an L2 portion that is "used to execute forwarding decision operations for frames forwarded among ports of the line cards." *Id.* 5:66-6:4. Similar to Smith and Ishimori, "[i]f the frame is received at the ingress card for the first time," the ingress forwarding engine learns the source MAC address of this frame. *Id.* 6:26-31. This involves "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* 6:31-34. Then, the ingress forwarding engine floods "copies of the fabric frame through its port interfaces to all (egress) line cards of the network switch," called the "flood-to-fabric (FF)" operation. *Id.* 6:34-39, 18:44-47.

A "novel MN [MAC notification] frame" complements the flooding operation. *Id.* 6:46-50. The MN frame "involves use of a primary input (PI) indicator," which "denotes a primary input MAC address that is directly attached to a port of the line card associated with the forwarding table containing this entry." *Id.* 6:50-56. As Edsall explains, the PI indicator is "asserted" for "a MAC address

that is learned from a frame sourced through one of the ports of the line card, as opposed to being learned through the switch fabric." *Id.* 6:56-60,18:56-19:5. The frame additionally includes a POE (port-of-exit) field that "includes a plurality of bits, one for each port interface of the switch fabric." *Id.* 6:24-25. The POE bit instructs the switch which port interfaces on which line cards should receive the MN frame. *Id.* 9:47-50.

2. Motivation to Combine Smith, Sharma, and Ishimori with Edsall

A POSITA would have been motivated to combine Smith, Sharma, Ishimori, and Edsall. EX1003 ¶¶105-109. All references disclose methods of using MAC forwarding tables in a distributed network switch. EX1004 ¶54; EX1026 3:42-50; EX1005 ¶9; EX1006 Abstract. It would have been obvious to a POSITA to modify the MAC forwarding tables, as disclosed in Smith, Sharma, or Ishimori, to implement the various features found in Edsall's forwarding engine. *Id.* ¶89. Specifically, a POSITA could alter the MAC tables to implement how Edsall's forwarding engine learns the source MAC address by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." EX1006 18:39-44; EX1003 ¶102.

A POSITA could further implement Edsall's "flood-to-fabric (FF) operations" by modifying the MAC tables to include POE bits that determine which port interfaces on which line cards should receive the MN frames. EX1006 18:47-

50. Finally, it would have been obvious to a POSITA to modify the MAC tables to include the PI indicator as taught in Edsall in order to keep track of whether a MAC address was learned from a local line card or a different line card. EX1003 ¶105. It would have also been obvious to a POSITA to further modify the MAC forwarding tables to implement the POE field in order to determine which ports should receive the MN frame. *Id.*

Additionally, Smith and Edsall are both assigned to the same assignee, and so a POSITA would have had a motivation to combine these references for this additional reason. *Abbot Vascular* at 28-29; *Laird Techs* at 30-31. Furthermore, creating entries in a MAC forwarding table and adding a PI indicator field and POE field to the MAC tables would have been a modification a POSITA would have known how to make with a reasonable expectation of success. EX1003 ¶109.

Ishimori additionally teaches an aging process using a "hit bit." EX1005 ¶9. The "hit bit" keeps track of how long a MAC address has been in the MAC address table without having been refreshed. *Id.* If the MAC address has not been refreshed within a predefined time, then the MAC address is deleted from the buffer on the next cycle. *Id.* "As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." *Id.* ¶25. A POSITA would have understood that MAC addresses may become stale due to changes in the network and would have been motivated to look at solutions such as Ishimori's aging parameter for

making sure the MAC table is up to date. EX1003 ¶108. Moreover, because the size of a MAC table is limited, a POSITA would have known older entries have to be aged to make space for newer entries. *Id.* A POSITA looking to implement this solution would have therefore been motivated to combine Smith and Ishimori and it would have been an easy application of Ishimori's methods to Smith's system. *Id.* For example, aging information could be added to the MAC address table described in Smith using known techniques of adding information to a table. *Id.*

Furthermore, one of skill in the art would have had a reasonable expectation of success in implementing this combination because it would have required a simple addition of Ishimori's aging process to Smith-Sharma-Edsall. *Id.* ¶104. Smith, Sharma, Ishimori, and Edsall already leverage link aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within the knowledge of a POSITA to simply implement the additional aging process taught by Ishimori to Smith-Sharma-Edsall, and a POSITA would have been able to do so with a reasonable expectation of success. *Id.*

3. Analysis of Ground 3

a) Claim 1[g]

To the extent this is not disclosed by the combination of Smith and Ishimori, Edsall discloses this. Edsall teaches the forwarding engine learns the *source* MAC address of a frame received at the ingress card for the first time. EX1006 18:39- 41.

It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* 18:42-44. "[I]f there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* 18:54-55. It would have been obvious to use Edsall's learning methods with the Smith-Sharma-Ishimori MAC tables, and a POSITA would have had a reasonable expectation of success in doing so. EX1003 ¶¶175-176.

b) Claim 1[h]

To the extent the Smith-Sharma-Ishimori combination does not render obvious this limitation, it would have been obvious in further view of Edsall. Edsall teaches the forwarding engine learns the *source* MAC address of a frame received at the ingress card for the first time. EX1006 18:39-41. Edsall further teaches "if there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* 18:54-55. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* 18:42-44. It then floods copies of the fabric frame to all the egress line cards of the network switch, which Edsall calls the "flood-to-fabric (FF) operation." *Id.* 18:47-50. This "forces each forwarding engine associated with each egress card to either (i) update its current L2 forwarding table entry with the newly-learned source MAC address and index of the frame or, if there is not a current entry, (ii) learn the source address/index of the frame." *Id.* 18:50- 55.

It would have been obvious to a POSITA to add Edsall's learning method and flood-to-fabric operation to the Smith-Sharma-Ishimori path learning operations. EX1003 ¶¶177-178. A POSITA would have been able to implement this teaching with a reasonable expectation of success because the combination of Smith and Ishimori already teach a message of new associations are sent to the plurality of member line cards. EX1004 ¶63; EX1005 ¶16. It would have been within the knowledge of a POSITA to create new entries in a FDB when an association of a MAC address and port does not yet exist. EX1003 ¶178.

c) Claim 4: The method according to claim 3 and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

Edsall discloses the "PI indicator" (e.g., marking the records) is asserted when a forwarding table entry for the MAC address is learned through one of the ports (e.g., data packets transmitted via a port of the line card) "as opposed to through the switch fabric" (e.g., received from another of the line cards). EX1006 6:46-64. Additionally, during prosecution, the Examiner found Edsall disclosed this limitation, which the applicant did not dispute. EX1002, 82,120-122. Specifically, Edsall discloses the "PI indicator is asserted for a destination MAC address entry of the forwarding table on the egress card and the DI contained in the switched fabric

frame (i.e., the ingress DI) is ***different*** from the DI stored in this egress forwarding table (i.e., the egress DI)." EX1006 18:56-19:5; EX1002, 82. Thus, the PI indicator is different for the MAC address entry learned through one of the ports as opposed to through the switch fabric. EX1006 6:46-64, 18:56-19:5; EX1003 ¶¶179-180.

d) Claim 5[a]: The method according to claim 4, and comprising: associating a respective aging time with each of the records;

Ishimori teaches each node has a hit bit that is set to "1" at the time of learning (e.g., associating a respective aging time with each of the records). EX1005 ¶9. The hit bit is then set to "0," so on the next aging process, "the corresponding MAC address is deleted from the buffer." *Id.* Thus, the hit bit is how Ishimori associates a respective aging time with each of the records. EX1003 ¶¶181-183. It would have been obvious to a POSITA to implement a "hit bit" on the Smith's lookup tables with Edsall's PI indicator in order to associate a respective aging time with each of the records. *Id.* ¶183.

e) Claim 5[b]: refreshing the records in the FDB responsively to further packets transmitted by the line cards; and

As described above, Ishimori teaches when the hit bit is "0," "the corresponding MAC address is deleted from the buffer." EX1005 ¶¶9, 11. Ishimori then teaches when the destination MAC address is received at card #0, the source MAC address is learned by "overwriting," and the hit bit is returned to "1" (e.g.,

refreshing the records in the FDB responsively to further packets transmitted by the line cards). *Id.* ¶11. It would have been obvious to a POSITA to implement Ishimori's "hit bit" on Smith's lookup tables with Edsall's PI indicator and to refresh the hit bit by learning the source MAC address of the data packets transmitted by the line cards. EX1003 ¶¶184-185.

f) Claim 5[c]: removing the records from the respective FDB if the records are not refreshed within the respective aging time.

Ishimori teaches if the hit bit remains "0" in the next aging process, then the corresponding MAC address is deleted (e.g., removing the records from the FDB if the records are not refreshed within the respective aging time). EX1005 ¶¶9,12. It would have been obvious to a POSITA implement Ishimori's "hit bit" on Smith's lookup tables with Edsall's PI indicator and to remove the records from the respective lookup tables if the record has not been refreshed by the next aging process. EX1003 ¶¶186-187.

g) Claim 6

Smith teaches sending a message that is a synchronization packet. §IX.A.4.k). Edsall teaches the "plurality of line cards" are "interconnected by a switch fabric 550," which could be the "switching core." EX1006 8:20-27. A POSITA would have understood that the packet is sent from the first line card via Edsall's "switching fabric" to at least the second line card. EX1003 ¶188.

h) Claim 7: The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed

Smith teaches its virtual network device includes several line cards, which include several interfaces. EX1004 ¶46; EX1003 ¶¶189-191. "[W]hen updating control protocol behavior of virtual link bundle 250(1), a user can simply access virtual network device sub-unit 122(1) (instead of accessing both virtual network device sub-units 122(1) and 122(2))." EX1004 ¶59. "Virtual network device sub-unit 122(1) can then automatically propagate to network device 122(2) any changes made by the user to the control protocols." *Id.* Smith teaches "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶62.

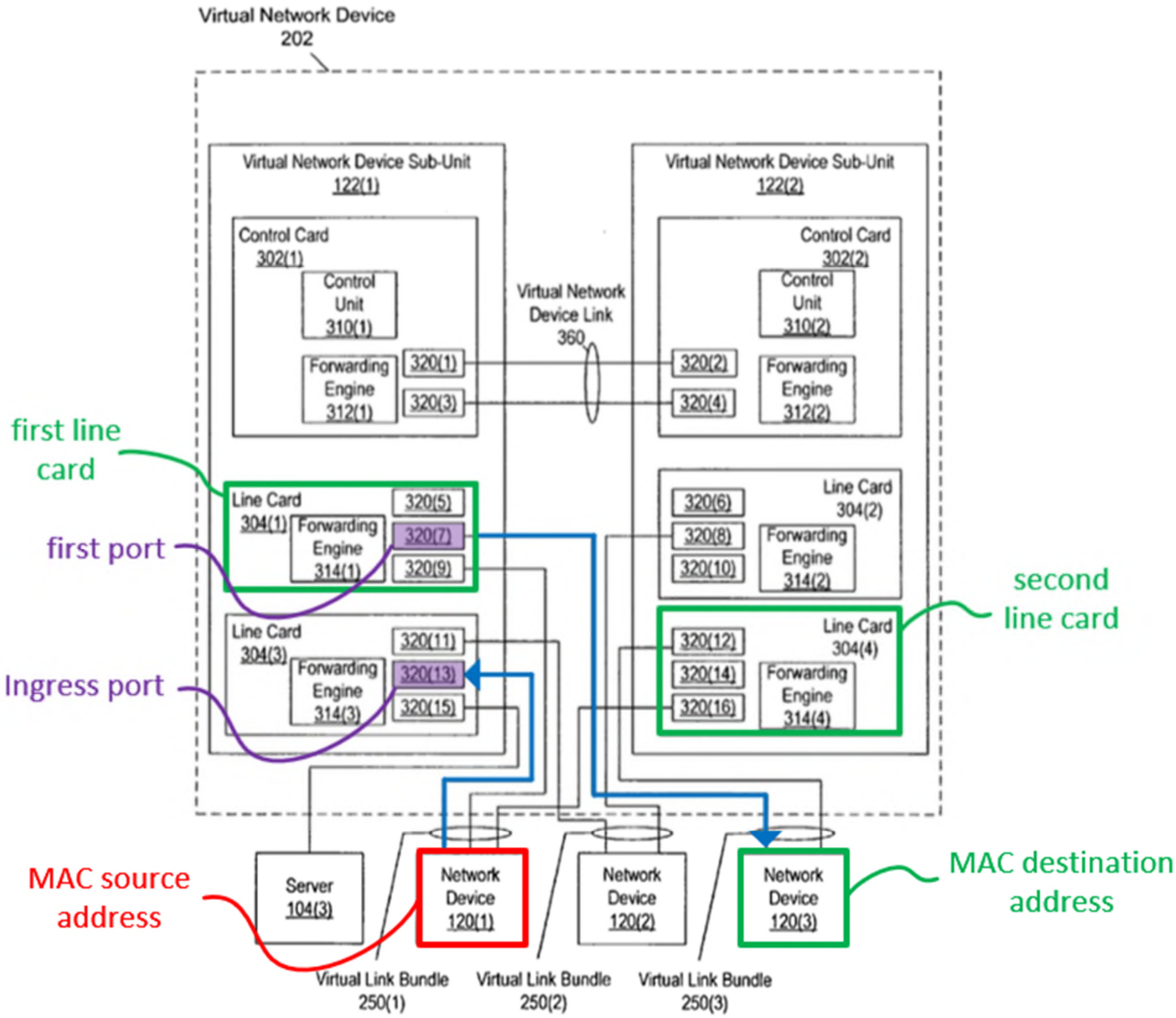


FIG. 3

Thus, if the record in the FDB of a line card (e.g., line card 304(3)) associates the MAC source address (e.g., virtual device 120(1)) with a port different from the one of the ports on which the data packet was received (e.g., a port other than 320(13)), the MAC notification frames will notify and update the L2 tables in the virtual network device 202, which would include at least the second line card (e.g., line card 304(4)) to indicate the record has been changed. Similarly, Ishimori

teaches the routes are transmitted via a "learn packet" (e.g., synchronization packet). EX1005 ¶25. During the packet transfer, each of the communication cards learns information about the route (e.g., the second line card). *Id.* ¶¶28,33; EX1003 ¶190.

Edsall further discloses the forwarding engine generates an MN frame (e.g., synchronization packet) that may get sent to the SMC (switch management card) to ensure that FwdT0 (e.g., the forwarding table) is synchronized. EX1006 17:26-38. The forwarding engine also asserts an appropriate bit of the POE field (port-of-exit field) when generating the MN frame, which is a port different from one of the ports on which data was received. *Id.* It would have been obvious to a POSITA to modify the MN frame from Smith to include the POE field in order to note the port interface of the switch fabric. *Id.* 6:24-25; EX1003 ¶191. This disclosure was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83,120-122.

i) Claim 10[a]: The method according to claim 1, and comprising: conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;

Smith teaches the uplink interface receives a data packet with the "sending device's MAC address" (e.g., a MAC source address). EX1004 ¶54; §IX.A.4.e); EX1003 ¶¶192-193. Based on Figure 3, a further data packet received from a further MAC source address (e.g., network device 120(3)) would be conveyed to the second line card (e.g., line card 304(4)) for transmission over the network.

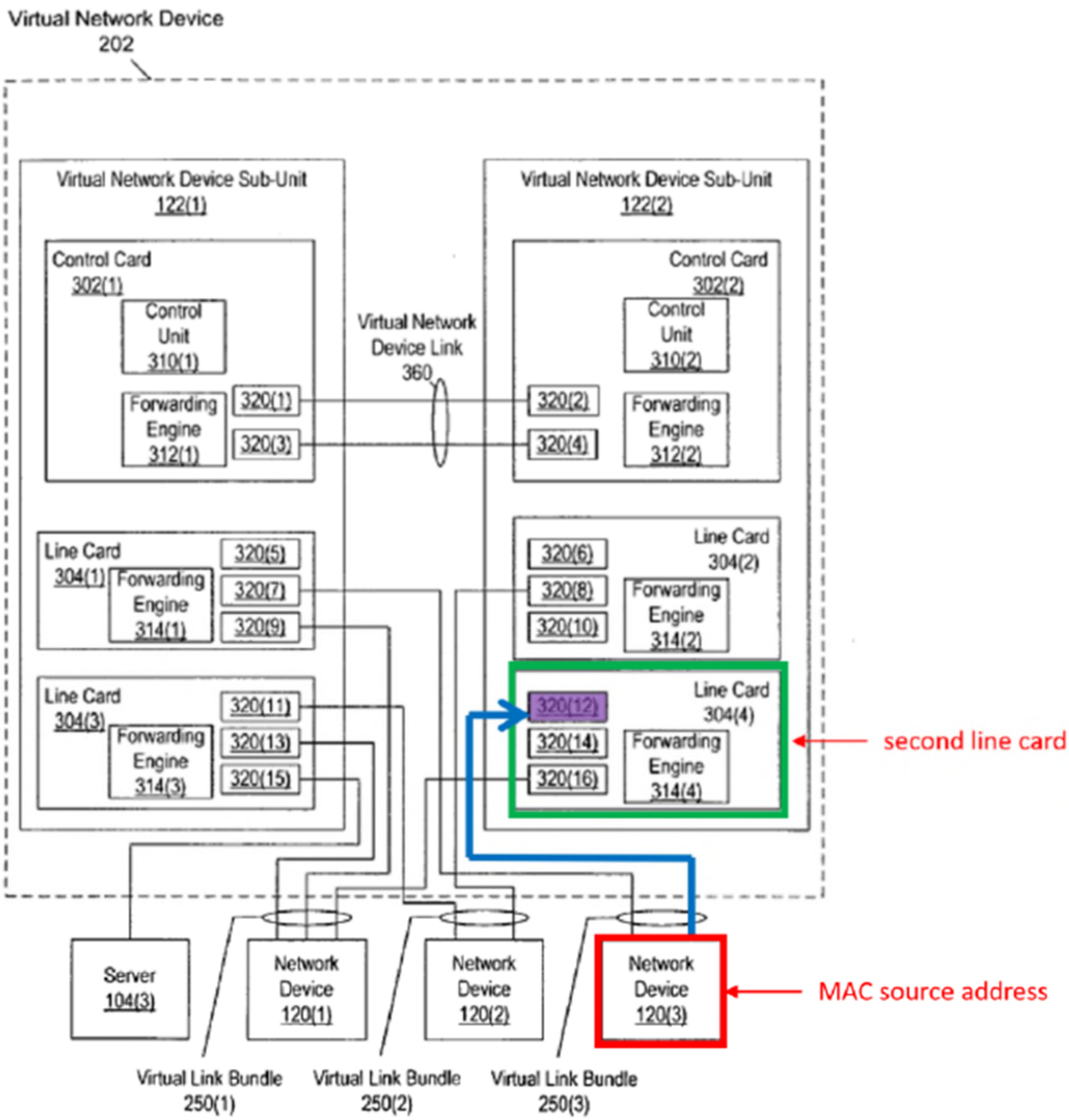


FIG. 3

To the extent Smith does not teach this claim, Edsall teaches in each subsequent frame, the encoded address recognition logic (EARL) circuit looks up the MAC address and "sends the corresponding rewrite information over the local bus after the frame" (e.g., conveying a further data packet to the second line card for

transmission over the network). EX1006 14:22-34. This was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83-84,120-122.

j) Claim 10[b]: checking the further MAC source address against the records in the FDB of the second line card; and

Smith teaches the virtual network device sub-unit learns the source identifier of the sending device (e.g., MAC source address). EX1004 ¶65; §VII.A.2.h), §VII.A.2.i); EX1003 ¶¶194-195. These identifiers are stored in a lookup table (e.g., FDB) on virtual network device sub-unit. EX1004 ¶61. The MAC source address (e.g., network device 120(3)) would be checked in the records of the FDB of said second line card (e.g., line card 304(4)). EX1003 ¶194.

To the extent Smith does not disclose this limitation, Edsall does. Edsall teaches checking to see if the rewrite information matches (e.g., checking the MAC source address against the records in the FDB of the second line card). EX1006 14:22-34 ("The destination port circuitry (or, alternatively, a UDlink or central rewrite engine) matches the frame with the rewrite information and modifies the frame as needed by replacing, *inter alia*, the destination and source MAC addresses."). This was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83-84,120-122.

k) Claim 10[c] responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and

sending a further message to inform at least the first line card of the further record.

Smith teaches the network device sub-unit 122(2) sends a MAC notification (e.g., a message) to update the forwarding engines (e.g., sending a further message to inform at least the first line card) when it learns of a new association (e.g., the further record). EX1004 ¶63; §IX.A.4.i); EX1003 ¶¶196-197. "After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." EX1004 ¶63. A POSITA would understand that the MAC notification (e.g., message) is therefore sent to inform at least the first member line card (e.g., line card 304(1)) of the new association. EX1003 ¶¶196-197.

To the extent Smith or Sharma does not disclose this limitation, Edsall teaches that the forwarding engine "modifies the frame as needed by replacing" the "destination and source MAC addresses" (e.g., adding a further record with respect to the MAC source address to the FDB). EX1006 14:22-34. This was cited by the examiner during prosecution and was not disputed by the applicant. EX1002, 83-84,120-122. It would have been obvious to a POSITA to use any of the learning methods from Smith, Sharma, Ishimori, or Edsall, which all disclose updating a second line card with a further record and sending a message to inform at least the first line card of the further record. EX1003 ¶197.

l) Claim 11[a]

To the extent this is not disclosed by the Smith-Sharma-Ishimori combination, Edsall discloses this. The ’400 patent describes the “switching core” as linking the multiple line cards. EX1001, 6:8-10. Edsall teaches the “plurality of line cards” are “interconnected by a switch fabric 550.” EX1006 8:20-27. Thus, a POSITA would have understood that the “switching fabric” in Edsall could be a “switching core.” EX1003 ¶198.

m) Claim 11[d]

See §§IX.A.4.p),IX.B.3.h),IX.C.3.a)-IX.C.3.b); EX1003 ¶199.

n) Claim 14: The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

See §IX.C.3.c); EX1003 ¶200.

o) Claim 15: The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.

See §§IX.C.3.d)-IX.C.3.f); EX1003 ¶201.

p) Claim 16

See §IX.C.3.g); EX1003 ¶202.

q) Claim 17: The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.

See §IX.C.3.h); EX1003 ¶203.

r) Claim 20: The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

See §§IX.C.3.i)-IX.C.3.k); EX1003 ¶204.

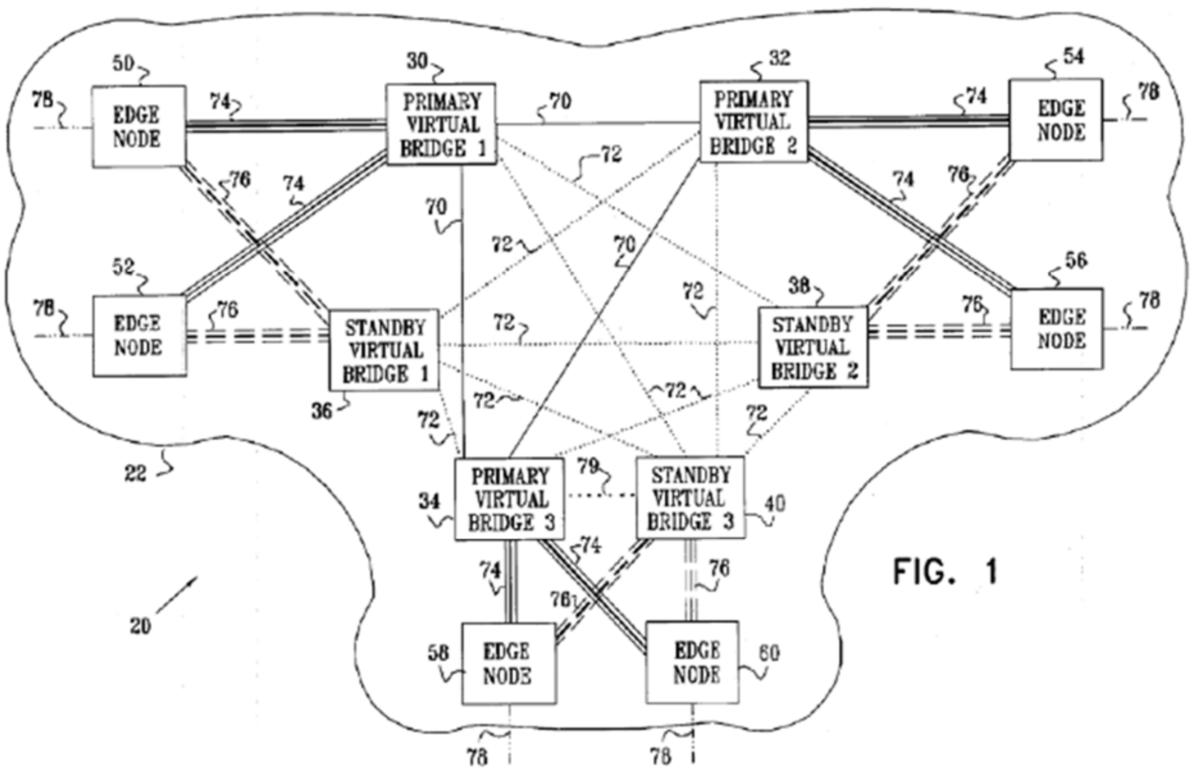
D. Ground 4: Claims 8-9 and 18-19 would have been obvious over the Smith-Sharma-Ishimori-Zelig combination

The combination of Smith, Sharma, Ishimori, and Zelig renders obvious claims 8-9 and 18-19.

1. Zelig

Zelig is directed to a data communication network that "includes a plurality of primary virtual bridges, interconnected by primary virtual connections." EX1007 Abstract. Zelig discloses "MAC bridges that implement the 802.1D standard allow

MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* ¶3. A MAC bridge "includes two or more MAC devices that interconnect the bridge ports to respective LANs." *Id.* The MAC bridges "maintain a database to map destination MAC address of the packets they receive to bridge ports." *Id.* ¶4. This database is built "by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* The purported invention in Zelig is directed to providing "improved mechanisms for protection from failure in virtual private networks (VPNs)." *Id.* ¶18. Zelig thus discloses a data communication network including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" that are "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶21-24. "[E]ach of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table." *Id.* ¶27. The VPN 20 includes multiple primary virtual bridges. *Id.* ¶42.



2. Motivation to combine Smith, Sharma, and Ishimori with Zelig

A POSITA would have been motivated to combine Smith-Sharma-Ishimori with Zelig. EX1003 ¶¶110-111. The Smith-Sharma-Ishimori combination would have been a virtual network device with several virtual network device sub-unit operating as a single logical network device with MAC bridges. EX1004 ¶54; EX1026 3:42-50; EX1005 ¶9; EX1007 Abstract. Zelig teaches that MAC bridges "maintain a database to map destination MAC address of the packets they receive to bridge ports." EX1007 ¶4. This database is built "by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* In order to "improve[] mechanisms for

protection from failure in virtual private networks (VPNs)," Zelig teaches a data communication network including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" can be "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶18,21-24.

A POSITA looking to prevent failure in the Smith-Sharma-Ishimori VPN would look to Zelig and implement Zelig's MAC bridges into the Smith-Sharma-Ishimori system to create a separate MAC table for each MAC bridge as taught in Zelig. EX1003 ¶110. It would have also been obvious to a POSITA to configure each virtual MAC bridge to serve a respective VPN, as shown above in Figure 1 of Zelig. *Id.* ¶96. Furthermore, implementing the MAC bridges as taught in Zelig to the Smith-Sharma-Ishimori data communication network would have been a modification that a POSITA would have known how to make with a reasonable expectation of success because, for example, Zelig teaches how to apply the techniques to systems having similar structure as Smith-Sharma-Ishimori. *Id.* ¶¶ 96,110.

3. Analysis of Ground 4

a) Claim 8: The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses

with the respective ports are maintained independently for each of the VPN instances.

Zelig discloses its VPN 20 contains multiple primary virtual bridges 30, 32, and 34 (e.g., multiple virtual MAC bridges in a Layer 2 VPN). EX1007 ¶¶42-43. It would have been obvious to a POSITA that the virtual bridges servicing VPN 20 are configured to serve that VPN. EX1003 ¶¶206-207. Zelig further teaches "each of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table . . . ." EX1007 ¶31. Thus, Zelig discloses each MAC bridge maintains its own MAC table, and because the MAC bridge serves only that VPN instance, the records associating the MAC addresses with the respective ports are maintained independently for each VPN instance. EX1003 ¶207.

b) Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

Smith teaches the MAC notification (e.g., message) is "distributed to any other forwarding engines within virtual network device sub-unit 122(2)." EX1004 ¶63. Zelig provides "VPN 20 is built around a virtual private LAN service (VPLS), operating within a network 22." EX1007 ¶42. It would have been obvious to a POSITA to adopt the notification scheme in Smith to the architecture in Zelig so the

MAC notification (e.g., message) is sent to all the line cards not just in the virtual network device sub-unit, but to all the line cards that serve the VPLS instances. EX1003 ¶¶208-209. Furthermore, the IEEE 802.1Q standards disclose a VLAN identifier (VID) as a twelve-bit field that "uniquely identif[ies] the VLAN to which the frame belongs." EX1008 §9.3.2.3. Thus, it would have been within the knowledge of a POSITA to identify the VPLS instance in the message, such as using the VID, in order to inform all the line cards in the VPLS. EX1003 ¶209.

c) Claim 18: The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

See §IX.D.3.a); EX1003 ¶210.

d) Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

See §IX.D.3.b); EX1003 ¶211.

E. Ground 5: Claims 9 and 19 would have been obvious over the Smith-Sharma-Ishimori-Zelig-802.1Q combination

The combination of Smith, Sharma, Ishimori, Zelig, and 802.1Q renders obvious claims 9 and 19.

1. 802.1Q

The 802.1Q standard defines the architecture for Virtual Bridged LANs, its services, protocols, and algorithms. EX1008, Abstract. It was part of the effort of the IEEE to standardize virtual LAN services in bridged LANs. EX1003 ¶97. The standard discloses a twelve-bit VLAN identifier field is used to "uniquely identify the VLAN to which the frame belongs." EX1008 §9.3.2.3. Such a standard would have been known and within the knowledge of a POSITA. EX1003 ¶97.

2. Motivation to combine Smith, Ishimori, and Zelig with 802.1Q

A POSITA would have been motivated to combine Smith-Sharma-Ishimori-Zelig with 802.1Q-1998. *Id.* ¶112. All references pertain to MAC bridges in a data communication network. EX1004 ¶54; EX1026 3:42-50; EX1005 ¶9; EX1007 Abstract; EX1008, Abstract. Moreover, Zelig expressly cites to the 802.1Q standard. EX1007 ¶12. A POSITA would have therefore been motivated to look to the teachings of 802.1Q in combination with Smith-Sharma-Ishimori-Zelig to modify the messages from Smith-Sharma-Ishimori-Zelig with a VLAN identifier as disclosed in 802.1Q in order to identify the VPLS instance. EX1003 ¶112. Additionally, a POSITA would have been motivated to improve the commercial applicability of the combined system by modifying the Smith-Sharma-Ishimori-Zelig system to be compliant with the 802.1Q standard. *Id.* Furthermore, modifying the Smith-Sharma-Ishimori-Zelig message to add the VLAN identifier would have

been a modification that a POSITA would have known how to make with a reasonable expectation of success. *Id.*

3. Analysis of Ground 5

a) Claim 9

See §IX.D.3.b); EX1003 ¶213.

b) Claim 19

See §§IX.D.3.b),IX.D.3.d); EX1003 ¶214.

X. PTAB DISCRETION SHOULD NOT PRECLUDE INSTITUTION

A. PTAB should not exercise its discretion to deny institution under *Fintiv*.

1. Factor 1: Institution will increase the likelihood of stay

The District Court regularly waits until an institution decision before deciding whether to stay a case and when IPRs are instituted regularly stays cases. *Targus Int'l LLC v. Victorinox Swiss Army, Inc.*, No. 20-cv-464-RGA, Dkt. 199 (D. Del. Jul. 14, 2021) (granting stay of case following institution after denying stay pre-institution). Thus, this factor weighs against discretionary denial.

2. Factor 2: District Court schedule

The Arista litigation does not have a scheduling order yet. The current average time-to-trial for the Delaware district court is 39 months, and this IPR proceeding will likely conclude before trial of the Arista litigation. EX1024. Accordingly, this factor weighs against discretionary denial. *See Sand Revolution II, LLC v. Continental Intermodal Grp.*, IPR2020-01393, Paper 24 (June 16, 2020).

3. Factor 3: Petitioner's investment in IPR outweighs forced investment in litigation to date

The Arista litigation is in early stages, and investment in it has been minimal. *See PEAG LLC v. Varta Microbattery GMBH*, IPR2020-01214, Paper 8, 17 (Jan. 6, 2021). Accordingly, this factor weighs against discretionary denial.

4. Factor 4: The Petition raises unique issues

Invalidity contentions have not been served in the parallel district court proceeding. Invalidity contentions in the Cisco litigation were served on Patent Owner on September 28, 2022. Instituting this proceeding will allow the Board to address the asserted prior art, including Sharma, which is not asserted in Cisco's and Dell's petition, EX1028, IPR2023-00370, and the issues will be narrowed in the co-pending litigations due to the estoppel provisions of 35 U.S.C. § 315(e)(2).

5. Factor 5: Whether the Petitioner and Defendants in the parallel litigation are the same party

Petitioner is a defendant in the co-pending litigation. While the *Fintiv* case indicates that a difference between the district court defendant and the petitioner may weigh against discretionary denial, nothing within the *Fintiv* case suggests that the same party between the proceedings weighs in favor of discretionary denial. *See HP Inc. v. Slingshot Printing LLC*, IPR2020-01084, Paper 13, 9 (Jan. 14, 2021). This factor is neutral and should not be a basis for denying institution.

6. Factor 6: Other circumstances support institution

The prior art presented in this Petition renders the Challenged Claims unpatentable as obvious. The merits of Petitioner' arguments are compelling because they lead to a strong conclusion that one or more claims are unpatentable by a preponderance of the evidence. This factor therefore weighs against discretionary denial.

Accordingly, the *Fintiv* factors are either neutral or weigh against discretionary denial. Because this Petition was filed before the statutory bar date, institution should not be denied on discretionary factors.

B. PTAB should not exercise its discretion to deny institution under *Becton* and *Advanced Bionics*

Institution should not be denied under the framework set forth in *Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8 (Dec. 15, 2017) (§III.C.5, ¶1, precedential) and *Advanced Bionics LLC v. MED-EL Elektromedizinische Gerate GmbH*, IPR2019-01469, Paper 6 (Feb. 13, 2020) (precedential). While Edsall was before the Examiner during prosecution, this Petition relies on Edsall only for limitations that were not disputed by the applicant. Moreover, Edsall serves as only a supplemental reference for the other references. Therefore, the arguments presented in this Petition are not the same as, or substantially similar to, arguments previously presented to the Office. Thus, the first prong of the *Advanced Bionics* framework is not met, and there is no need to reach

the second prong to resolve against discretionary denial under §325(d). *Oticon Med. AB v. Cochlear Ltd.*, IPR2019-00975 Paper 15, 20 (Oct. 16, 2019) (precedential).

C. Discretionary denial under *General Plastic* is not appropriate

None of the *General Plastic* discretionary institution factors favor denial. *General Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 16 (Sept. 6, 2016) (Section II.B.4.i. precedential).

1. Factors 1-2

Factors 1 and 2 weigh against discretionary denial because the earlier-filed petition does not involve Arista.

Additionally, petitioners of the IPR2023-00370 certified that "[n]o unnamed entity is funding, controlling, or direction this Petition, or otherwise has had an opportunity to control or direct this Petition or Petitioners' participation in any resulting IPR." IPR2023-00370, Paper 1, at 2. Arista has not communicated with Dell or Cisco regarding the substance of their petition and had no notice of filing their petition until after it was filed. This further weighs against discretionary denial.

2. Factors 3-5

Given the short time between the filing of Cisco's and Dell's petition and the present Petition, Factor 3-5 also weigh against discretionary denial. *See Mercedes-Benz USA, LLC v. Carucel Investments, L.P.*, IPR2019-01404, Paper 12 at 13 (Jan. 22, 2020). This Petition is being filed less than four months after Cisco's and Dell's petition.

Importantly, Corrigent has not filed a Patent Owner Preliminary Response in IPR2023-00370, so Petitioner receives no potential benefit or tactical advantage. Factors 3-5 therefore weigh against discretionary denial.

3. Factors 6-7

There is substantial overlap between this Petition and Cisco's and Dell's petition challenging the same claims and using overlapping prior art. This Petition includes the same grounds as the Cisco petition with the addition of a single prior art reference for each of the grounds. The similarities that exist between these petitions provide strong basis for efficiency. The two petitions could be joined or consolidated, due to the substantial overlap in subject matter between the proceedings, and Arista agrees to allow the counsel for either Dell or Cisco to take the lead on overlapping issues.

Petitioner also has a meaningful due process interest to be heard in this forum on the merits, because of their choice to raise the unpatentability grounds in this Petition before the Board. This fact also favors not denying institution.

XI. CONCLUSION

Petitioner have established a reasonable likelihood that the Challenged Claims are unpatentable. Petitioner therefore respectfully request that IPR be instituted.

DATED: April 3, 2023

Respectfully Submitted,

/s/ Eliot D. Williams
Eliot D. Williams (Reg. No. 50,822)
Baker Botts L.L.P.
700 K Street, N.W.
Washington, D.C., 20001-5692
Tel: 202-639-1334
Eliot.Williams@BakerBotts.com

Attorney for Petitioner Arista Networks, Inc.

CERTIFICATION UNDER 37 C.F.R. § 42.24(d)

Under the provisions of 37 C.F.R. § 42.24(d), the undersigned attorney hereby certifies that the word count for Sections I and III-VIII of the foregoing Petition for *Inter Partes* Review is 13,616, according to the word count tool in Microsoft Word.

DATED: April 3, 2023

Respectfully Submitted,

/s/ Eliot D. Williams
Eliot D. Williams (Reg. No. 50,822)
Baker Botts L.L.P.
700 K Street, N.W.
Washington, D.C., 20001-5692
Tel: 202-639-1334
Eliot.Williams@BakerBotts.com

Attorney for Petitioner Arista Networks, Inc.

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a), (b) on the Patent Owner via US Express Mail and FedEx Priority Overnight of a copy of this Petition for *Inter Partes* Review and supporting materials at the correspondence address of record for the ’400 patent shown in USPTO PAIR:

May Patents Ltd. (attorney/agent of record)
c/o Dorit Shem-Tov
P.O. Box 7230
Ramat-Gan, - 5217102
Israel

Corrigent Corporation (assignee of record)
126 Yigal Allon Street
Tel Aviv, Israel 67443

Corrigent Corporation (assignee of record)
291 Main St.
C/O New England Intellectual Property
West Newbury, Massachusetts 01985

DATED: April 3, 2023

Respectfully Submitted,

/s/ Eliot D. Williams
Eliot D. Williams (Reg. No. 50,822)
Baker Botts L.L.P.
700 K Street, N.W.
Washington, D.C., 20001-5692
Tel: 202-639-1334
Eliot.Williams@BakerBotts.com

Attorney for Petitioner Arista Networks, Inc.

APPENDIX: CHALLENGED CLAIM LISTING

No.	Limitation
1[pre]	A method for communication, comprising:
1[a]	configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;
1[b]	configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards;
1[c]	providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;
1[d]	receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;
1[e]	conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;
1[f]	if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;
1[g]	checking said MAC source address of the data packet against records in said FDB of said first line card; and
1[h]	if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card of said plurality of member line cards.
2	The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
3	The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

No.	Limitation
4	The method according to claim 3, and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
5[pre]	The method according to claim 4, and comprising
5[a]	associating a respective aging time with each of the records;
5[b]	refreshing the records in the FDB responsively to further packets transmitted by the line cards; and
5[c]	removing the records from the respective FDB if the records are not refreshed within the respective aging time.
6	The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via a switching core of the network node to at least the second line card
7	The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.
8	The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances
9	The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.
10[pre]	The method according to claim 1, and comprising:
10[a]	conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;

No.	Limitation
10[b]	checking the further MAC source address against the records in the FDB of the second line card; and
10[c]	responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.
11[pre]	A node for network communication, comprising:
11[a]	a switching core;
11[b]	a plurality of ports;
11[c]	a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards,
11[d]	wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

No.	Limitation
12	The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
13	The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.
14	The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
15	The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.
16	The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.
17	The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.
18	The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN

No.	Limitation
	instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.
19	The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.
20	The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

EXHIBIT 3D

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ARISTA NETWORKS, INC.,
Petitioner

v.

CORRIGENT CORPORATION,
Patent Owner.

Case No. IPR2023-00805

U.S. Patent No. 7,593,400

DECLARATION OF TAL LAVIAN, PH.D., UNDER C.F.R. § 1.68 IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

Page

TABLE OF APPENDICES

Document	Description
Appendix A	*Curriculum Vitae*
Appendix B	Text of Challenged Claims

TABLE OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 7,593,400 (“the ’400 patent”)
1002	Copy of Prosecution History of the ’400 patent
1004	U.S. Patent Application Publication No. 2005/0198371 (“Smith”)
1005	Certified English Translation of Japanese Patent Application No. 2005086668 (“Ishimori”)
1006	U.S. Patent No. 6,735,198 (“Edsall”)
1007	U.S. Patent Application Publication No. 2004/0133619 (“Zelig”)
1008	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 1998 Edition (“802.1Q-1998”)
1015	IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition
1016	IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition
1017	IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition
1018	Kompella et al., *Virtual Private LAN Service*, IETF (December 2005)
1019	Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005)
1020	Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005)
1021	U.S. Patent No. 6,917,986
1022	U.S. Patent No. 7,974,223
1025	Japanese Patent Application No. 2005086668 (“Ishimori”)

1026	U.S. Patent No. 6,999,418 ("Sharma")
1027	*Corrigent Corp. v. Cisco Systems, Inc.*, No. 6:22-cv-00396 (W.D. Tex.), Dkt. 69
1028	*Dell Technologies Inc. et al v. Corrigent Corporation*, IPR2023-00370, Paper 1 (Dec. 23, 2022)
1029	Declaration of Nicholas Bambos, Ph.D. under 37 C.F.R. § 1.68

I, Tal Lavian, Ph.D., do hereby declare as follows:

I. INTRODUCTION

1. My name is Dr. Tal Lavian. I understand that I am submitting a declaration in connection with *inter partes* review ("IPR") proceedings before the United States Patent and Trademark Office for U.S. Patent No. 7,593,400 ("the '400 patent").

2. I have been retained on behalf of Arista Networks, Inc. ("Arista" or "Petitioner") to offer technical opinions with respect to the '400 Patent and the prior art relied on to show obviousness. I have been asked to provide my opinions as to whether the inventions claimed in claims 1-20 ("the Challenged Claims") of the '400 patent would have been obvious to a person with ordinary skill in the art ("POSITA") at the time of the alleged invention, considering the prior art.

3. I am being compensated for my work in this matter at my standard hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

II. QUALIFICATIONS AND EXPERIENCE

4. Based on my qualifications, education, knowledge, expertise, and experience, I believe I am qualified to offer opinions relating to the technology described in the '400 patent. My qualifications for forming the opinions outlined in

this declaration are summarized here and explained in more detail in my curriculum vitae. Appendix B. Appendix B also contains a list of my publications and patents.

5. I am the Principal Scientist for TelecommNet Engineering, Inc. I have been with the company since 2008. In my role at TelecommNet, I provide consulting and expert services in network communications, telecommunications, internet protocols, and smartphone mobile wireless devices. At TelecommNet, I also provide system architecture and technology analysis for computer networks, mobile wireless devices, and web-technology projects.

6. Further, I am the CEO and CTO of Aybell (previously VisuMenu, Inc.). I founded VisuMenu, Inc. in 2010. At VisuMenu, Inc., I led the software design and development of a visual interactive voice response system for smartphones and mobile devices based on innovative wireless and network communications technologies.

7. In 2016, VisuMenu, Inc. was rebranded as Aybell. At Aybell, I have facilitated the design, architectural development, and implementation of a cloud data center for connecting any smartphone user to any company and service by digitizing interactive voice systems and exposing them through cloud-service application programming interfaces to other applications.

8. In 2008, I was a Network Communications Consultant for Ixia, a computer and wireless networking company. At Ixia, I researched and developed advanced network communications technologies.

9. From 1996 to 2007, I held several roles for Nortel Networks, a telecommunications and networking equipment company. I was a Principal Scientist, Principal Architect, Principal Engineer, and Senior Software Engineer. And I was the Principal Investigator for the U.S. Department of Defense Projects. In this role, I conceived, proposed, and completed three research projects—active networks, DWDM-RAM, and a networking computation project for the Air Force Research Lab.

10. At Nortel Networks, I was also an Academic and Industrial Researcher. In this role, I designed software for switches, routers, and network communications devices and developed systems and architectures for switches, routers, and network management.

11. From 1987 to 1995, I worked for three voice and data communication and software/hardware companies—Aptel Communications, Scitex Ltd., and Shalev. In these roles, I developed a mobile wireless device, designed and managed a personal communication network and personal communication system, invented and implemented a two-way paging product, developed system and network

communications in C/C++, and developed real-time software and algorithms in C/C++ and Pascal.

12. Further, I have served as an industry fellow and lecturer at the UC Berkeley College of Engineering, Sutardja Center for Entrepreneurship. I have co-authored over 25 scientific publications, journal articles, and peer-reviewed papers.

13. I am a member of several professional organizations, including the Association of Computing Machinery ("ACM") and the Institute of Electrical and Electronics Engineers ("IEEE") (senior member). I am also certified under the IEEE Wireless Communications Engineering Technologies ("WCET") 2012 Program, specifically designed by the IEEE Communications Society ("ComSoc") to address the worldwide wireless industry's growing and ever-evolving need for qualified communications professionals.

14. I received a B.S. degree in Mathematics and Computer Science from Tel Aviv University in 1987, an M.S. degree in Electrical Engineering from Tel Aviv University in 1996, and a Ph.D. in Computer Science specializing in network communications from UC Berkeley in 2006.

15. I have testified in Federal courts, the PTAB, and the ITC on behalf of leading companies such as Amazon, LinkedIn, AT&T, Sprint, Cisco Systems, Juniper Networks, ZTE, Huawei, Motorola, HP, LG, Microsoft, Facebook, and Apple. I testified in over 80 depositions on network communications and

telecommunications, including telecommunications, network communications, network architecture, communications design, Internet protocols, and technologies. Furthermore, I have served as an expert witness in over 120 patent-related cases.

16. I am an inventor of over 120 patents, 60 of which I prosecuted pro se in front of the USPTO. A number of these issued patents are relevant or contemporaneous to the '400 patent.

17. I have published many peer-reviewed articles and publications on network technology, a relevant sampling of which includes: "Implementation of a Quality of Service Feedback Control Loop on Programmable Routers" at the 12th IEEE International Conference on Networks 2004, "Practical Active Network Services within Content-Aware Gateways" at the proceeding of the DARPA Active Networks Conference and Exposition, 2002, and "Active Networking on a Programmable Network Platform" at the Fourth IEEE Conference on Open Architectures and Network Programming.

18. I have spent over a decade as an academic and industrial researcher. I have worked researching and developing many projects during that time, partly through heading research collaboration with leading universities and professors at UC Berkeley, Northwestern University, University of Amsterdam, and the University of Technology, Sydney. Some of these projects include Data-Center Communications: network and server orchestration, Network resource orchestration

for Web services workflows, and Packet capturing and forwarding service on IP and Ethernet.

19. I am a member of several professional organizations, including the IEEE Communications Society (COMMSOC), the ACM Special Interest Group on Data Communication (SIGCOM), the IEEE Consultants' Network (CNSV), and the Global Member of the Internet Society (ISOC).

20. I have many accomplishments in the field of network technology, such as leading the development of the first network resource scheduling service for grid computing, managing and engineering the first demonstrated dynamic transatlantic allocation of 10Gbs Lambdas as a grid service, and successful demonstration of the first wire-speed active network on commercial hardware.

21. Based on my above-described over three decades of experience in network communications techniques, network architectures, network topologies, quality of service, network resource allocation, and the acceptance of my publications and professional recognition by societies in my field, I believe that I am qualified to be an expert in the field of network technology.

22. Based on my knowledge, education, training, experience, and expertise described above, and as indicated in my curriculum vitae, I am qualified to provide the following opinions with respect to the patents in this case. Additionally, I am at

least a person having ordinary skills in the art as of the priority date of the ’400 Patent.

23. My *curriculum vitae* contains further details on my education, experience, publications, and other qualifications to render an expert option. I am being compensated at my standard consulting rate for my work on this declaration. My compensation is not dependent on the outcome of this case, and I have no financial interest in the outcome.

III. PERSON OF ORDINARY SKILL IN THE ART

24. In evaluating the prior art references and other material, I have used the perspective of a person of ordinary skill in the art (“POSITA”) to which the patent is related as of the time of the patent’s priority date. I am informed by Counsel that a POSITA is presumed to be aware of pertinent prior art and the conventional wisdom in the art and is a person of ordinary creativity. I have applied this standard in my analysis throughout my declaration.

25. The ’400 patent is entitled “MAC Address Learning in a Distributed Bridge” and is directed to the field of communication networks. ’400 patent at 1:6–7. A POSITA in the 2006 timeframe, which I am informed by Counsel is the earliest claimed effective filing date of the ’400 patent, would have had at least 1) a bachelor’s degree in electrical engineering, computer engineering/science, or a related field, and 2) either (a) a master’s degree in electrical engineering, computer

engineering/science, or a related field, or (b) two or more years of work or research experience in networking and computing; that is, more education can compensate for less professional experience.

26. I meet the criteria described above and I am a person with at least ordinary skill in the art pertaining to the '400 patent. I would have qualified as such a person by at least 2006, which I understand is the earliest priority date of the '400 patent.

IV. APPLIED LEGAL PRINCIPLES

27. I am informed by Counsel for the Petitioners about the following legal standards, which I have applied throughout my analysis.[1]

[1] I am informed by Counsel that the patent laws were amended by the America Invents Act (AIA), but that the earlier statutory requirements still apply to pre-AIA patents. I have been informed by Counsel that the '400 patent is a pre-AIA patent, so the pre-AIA requirements control. Unless otherwise stated, my understanding of the law about patent invalidity as set forth in this Declaration relates to the pre-AIA requirements.

A. Prior Art

28. I am informed by Counsel that a patent, published patent application, or other publication, must first qualify as prior art before it can be used to invalidate a patent claim.

29. I am further informed by Counsel that a U.S. or foreign patent qualifies as prior art to a patent claim if the date of application, issuance, and/or publication of the prior art patent is prior to the purported date of invention of the patent claim.

30. I am additionally informed by Counsel that a printed publication, such as a technical publication, a magazine article, or newsgroup post, qualifies as prior art to a patent claim so long as the date of publication is prior to the purported date of invention of the patent claim.

31. I am moreover informed by Counsel that a U.S. patent qualifies as prior art to a patent claim if the application for the prior art patent was filed in the United States before the purported date of invention of the patent claim.

32. I am further informed by Counsel that, if a U.S. Patent incorporates another patent or document by reference, it is considered to be part of that same reference.

B. Anticipation

33. I am informed by Counsel that, once the claims of a patent have been construed, the second step in determining anticipation of a patent claim requires a

comparison of the construed claim language to the prior art on a limitation-by-limitation basis.

34. I am further informed by Counsel that a prior art reference anticipates a claim, and thus renders the claim invalid, if all elements of the claimed process, system, or device are disclosed in the prior art reference, either explicitly or inherently.

35. I am also informed by Counsel that, under Section 102 of the Patent Act, claims of a patent are invalid for lack of novelty if they are anticipated by a single prior art reference. I am further informed by Counsel that a claimed invention is invalid as anticipated and hence lacks novelty if all of the limitations of the claim as construed by the Court are present in a prior art reference, or any document that the reference incorporates by reference. However, I am informed by Counsel that a limitation of the claim need not be shown directly in a reference so long as a POSITA would have understood the limitation to be inherent, or necessarily present in the reference.

C. Obviousness

36. I am informed by Counsel that an invention may also be obvious, even though the invention is not identically disclosed or described in a single prior art reference, which is required for anticipation. This is true if the differences between the subject matter sought to be patented and the prior art are such that the subject

matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the relevant art.

37. I am further informed by Counsel the Supreme Court has urged caution in granting a patent based on the combination of elements found in the prior art. This is because a patent for a combination that only unites old elements with no change in their respective functions withdraws what already is known from the public, giving it exclusively to the patentee. The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.

38. I am also informed by Counsel that when a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, the invention is likely obvious. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill. Put another way, I am informed by Counsel that a court must ask whether the improvement is more than the predictable use of prior art elements according to their established functions. I am further informed by Counsel that this is a flexible standard and it is not confined to any particular formalistic conception.

39. I am also additionally informed by Counsel that, when conducting an obviousness inquiry, the analysis should consider the scope and content of the prior art, differences between the prior art and the claims of the challenged patent claims, and the level of ordinary skill in the art. Secondary considerations of nonobviousness can also be considered, which include commercial success enjoyed by devices practicing the patented invention, industry praise for the patented invention, copying by others, and the existence of a long-felt but unsatisfied need for the invention. I am also informed by Counsel that there must be a nexus between these factors and the claims themselves for these secondary considerations to impact the analysis of obviousness.

40. I am informed by Counsel that there generally must be a motivation to combine the references in order to find obviousness. This motivation or reason to combine can arise from the references themselves, changing market dynamics, or from simple common sense.

41. I am further informed by Counsel that a single reference can render a patent claim obvious, even if that reference does not fully anticipate the claim. To determine whether there was an apparent reason to combine the known elements in the way a patent claims, it will often be necessary to look to interrelated teachings of multiple patents; to the effects of demands known to the design community or

present in the marketplace; and to the background knowledge possessed by a person having ordinary skill in the art.

D. Claim Construction Standard

42. I am informed by Counsel that, during *inter partes* review, the same standard is used as during claim construction in district court civil actions. I am further informed by Counsel that, in the district court context and in this IPR, the words of a claim are generally given their ordinary and customary meaning. This is the meaning the term would have to a person of ordinary skill in the art in question at the time of the invention, which generally means the time the application was filed.

43. I am further informed by Counsel that the specification informs the proper interpretation of a claim. This is because a person of ordinary skill in the art is deemed to have read the claim term not only in the context of the particular claim in which the dispute term appears, but in the context of the entire patent, including the specification. I am also informed by Counsel that the prosecution history and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art are also considered in ascertaining the meaning of claim terms.

V. TECHNOLOGY BACKGROUND

44. To put in context my opinions on the validity of the claims of the '400 patent, in this section I provide some background on computer networks. All of the concepts discussed in this section were well-known and widely used well before the May 19, 2006, priority date for the '400 patent.

A. Overview of Computer Networks

45. Computer communication networks are comprised of communication links connected to ports of communication nodes (*e.g.*, switches, routers, computer hosts) forming a graph of (communication) nodes and (communication link) edges. Information communication between nodes occurs in data units called packets (or frames in some cases), which are finite sequences of 0/1 bits representing information in digital form. In general, each packet includes a destination address, that is, the identifier (ID) of the final/destination node that it wants to reach. Each intermediate node on the path of a packet (i.e., the sequence of links and nodes it traverses) from its source (i.e., the node that injected it into the network) to its destination receives the incoming packet via an input port from an upstream node and retransmits it via an output port to a downstream node, which typically depends on the destination address of the packet.

46. A rough analogy is that of viewing the packet as a "car," the communication network as a "freeway system," the communication nodes as

"freeway intersections" and the communication links as the "freeway segments" connecting the intersections. When the "car" reaches a "freeway intersection" on a "freeway segment," the freeway signs direct (route) the car onto another "freeway segment" towards another "freeway intersection" and so on, until it reaches its destination and exits on a freeway ramp.

47. The path of a packet through a network, from its source node to its destination node, should have no loops under normal network operation. Clearly, a packet going around in loops and visiting any node more than once creates many problems, including 1) increasing the delay to deliver the packet to its destination (since it has to lose time to traverse a loop of links and nodes to end up back at the node it entered the loop from) and 2) misusing the communication bandwidth of the nodes and links on the loop and, hence, causing unnecessary congestion on the network and delaying also other packets that are traveling though the network. Therefore, it is important that the link/node path traversed by each packet on the network be loop-free under normal network operation. The rough analogy of the communication network to the "freeway system" also makes it clear why we should avoid loops; a "car" should not loop around and visit the same "freeway segment" and/or "freeway intersection" twice at any point while traveling towards its destination under normal "freeway" conditions.

48. A tree is a graph where there is only one path (i.e., sequence of consecutive links/edges) to go from any node to any other one, as opposed to having more than one distinct path to go from some node to some other node. (i.e., a tree is a loop-free graph). Given any general graph (e.g., even with loops), a spanning tree of the graph is a tree (i.e., loop-free) subgraph that includes all nodes of the original graph, but possibly fewer links in order to break any loops that may exist in the original graph and reduce the original graph to a tree, hence, allowing no loops.

49. Spanning tree algorithms are used to construct spanning trees of communication network graphs in order to route packets on loop-free paths. Such algorithms are embedded in the Spanning Tree Protocol (STP) which runs on computer communication networks in order to detect and remove loops, so that packets can be forwarded to their destinations on loop-free paths.

B. Switches, Learning MAC Tables, Spanning Trees

50. The Open Systems Interconnection (OSI) reference model is a layered modular abstraction of computer communication networks, especially with respect to network protocols (*i.e.*, operational schemes or mechanisms).

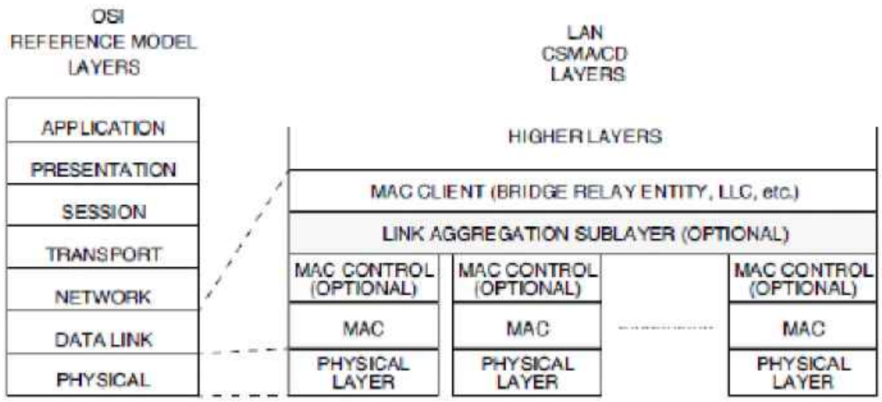


Figure 43–1—Architectural positioning of Link Aggregation sublayer

(Ex. 1017, Section 3, Fig. 43-1).

51. Layer 1 (L1) is the Physical layer, layer 2 (L2) is the Data Link layer and layer 3 (L3) is the Network layer; on top of those layers others are also defined. A sub-layer of the Data Link layer L2 is the Media Access Control (MAC) one. For example, a widely used computer communication (family of) protocol(s) generally conforming to the L1, L2, L3 layers of the OSI model is the Ethernet (family of) protocol(s), which has been standardized by IEEE under the name 802.3.

52. Each network port/interface device on a network node has a unique hardware identifier (ID), which is called "MAC address" and is generally inserted into the device by its manufacturer. MAC addresses are used by L2 switches (also called bridges historically) to switch packets at network nodes from input ports to output ports. L2/MAC switches/bridges can interconnect separate physical LANs into a unified logical network.

53. L2 switching nodes, including in the context of Ethernet IEEE 802.3 technology, operate at the L2/MAC layer to allow transfer of data packets called "frames" from switch input ports to output ports. Each frame includes the MAC address of its source node and the MAC address of its destination node. Each node maintains a MAC table, which specifies at which output port the switch should output an incoming/received frame, based on its MAC destination address. That is, when the switch receives a frame, it "looks" at the frame's MAC destination address and "consults" its MAC table to find the port from where it should next transmit the frame to continue on the path to its destination. Thus, each switch's MAC table maps the incoming frames' MAC destination addresses to the switch's output ports, in order to transmit each received frame on to the downstream node on its path to its destination.

54. The MAC tables are "learned" dynamically and distributedly. Indeed, when an incoming frame is received at a port of the switch, the latter "looks" at the source MAC address of the frame, and infers and "learns" that this MAC address is reachable through that port, and "keeps" that information in its MAC table. Thus, the MAC table is (backward) "learned" by the switch, in the sense that if the switch sees an incoming frame with a certain source MAC address at one of its ports, then it infers and "learns" that this MAC address is reachable from that port (as a destination MAC address now). Therefore, next time the switch receives a frame

with that MAC address as a destination MAC address, it will output the frame on that port.

55. When a switch is first connected to the network in a "plug-and-play" manner, its MAC table is empty and the switch "knows nothing" – it has to autonomously "learn as it goes." As mentioned before, when it receives an incoming frame at a port it "learns" that the source MAC address of the frame is also reachable as a destination MAC address from that port and "records" this information in its MAC table (if it hasn't done so already with a previously received such frame). When the switch receives a frame at one of its ports, the switch looks at the destination MAC address of the frame and consults its MAC table: 1) If it finds an entry that this destination MAC address is reachable from one of its ports, then it transmits the frame on that port. 2) If it does not find such an entry, the switch transmits the frame on all its ports (floods it to its ports), except on the one where it received the frame from. As the switch operates, its MAC table is populated by more and more information entries mapping destination MAC addresses to switch ports from where these destination MAC addresses are reachable. After a while the MAC table of the switch has been built up enough to give it visibility into the network to forward frames to their destinations efficiently.

56. Of course, in general there may be multiple redundant links and paths to get from a source to a destination port on a network of switches for at least

purposes of increasing reliability and robustness to component failures. Such a topology may include path loops which can cause frames to go in circles. Besides the detrimental effects mentioned above, allowing frames to go in loops is additionally detrimental to network "stability" in various ways.

57. For example, suppose that switch A is connected to switch B via two links, the first one connecting port A1 of A to port B1 of B, and the second connecting port A2 of A to port B2 of B. When a frame with a previously unobserved (by A and B) destination MAC address arrives to switch A on a port A3, it is output on both ports A1 and A2 to B1 and B2 respectively. Since the destination MAC address is also previously unobserved by switch B, when the frame is received by B1 it will be output on B2 (to A2) and also the (same) frame received on B2 will be output on B1 (to A1), and so on and so forth, which could cause an infinite retransmission loop of the frame. To prevent formation of path loops, the switch network runs a Spanning Tree Protocol, which identifies redundant links and blocks transmission on them, leaving a single path to go from any source to any destination on the switch network.

58. Component failures, like port/link failures or whole switch/node failures, or even planned connection of new components to the network, or additionally movement (disconnection and reconnection) of existing components to other parts of the network cause reconfiguration of the whole network. Therefore,

1) the Spanning Tree Protocol will have to be rerun on the switches to identify a new spanning tree to forward frames from their sources to their destinations and 2) at least the MAC table entries corresponding to the topology changes will have to be purged or flushed and then "relearned" on the switches, as the old ones do not map to the new network topology any more.

C. Link Bundling and Link Aggregation

59. Multiple links (and their corresponding ports) connecting switches can be "bundled together," for example, into a Link Aggregation Group in some cases for various reasons, including increased bandwidth, traffic load balancing, scalability, increased reliability, etc. These Link Aggregation Groups are treated as a single logical link. For example, the IEEE 802.3 standard (as shown in Fig. 43-1 of Ex. 1017, Section 3, reproduced above) shows a Link Aggregation Sublayer of Layer 2 (*i.e.*, the Data Link layer of the OSI model), which can combine a number of individual physical links into a single logical link and present a single MAC interface to the MAC client.

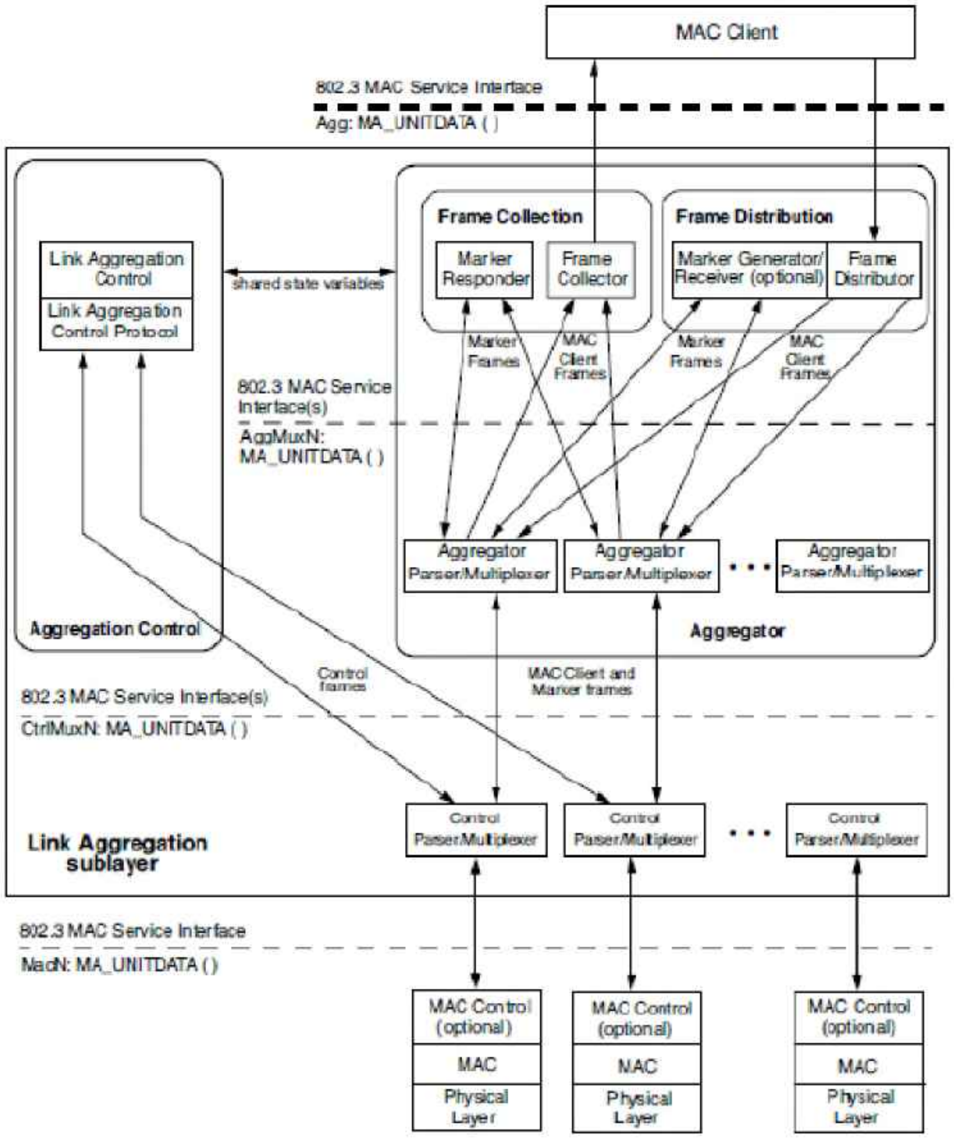


Figure 43–2—Link Aggregation sublayer block diagram

60. The architecture of this sublayer is shown in Fig. 43-2 of Ex. 1017 (IEEE 802.3 Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications), Section 3 reproduced above showing the Frame Collection/Distribution modules and the aggregation control module, including the Link Aggregation Control Protocol (LACP) module.

61. "Link bundles" or LAGs across multiple physical switches would provide redundancy by eliminating a single point of failure: if one of the link/switches in the "bundle" or LAG went down, the other links/switches could still operate.

VI. BACKGROUND OF THE '400 PATENT

A. Summary of the '400 Patent

62. The '400 patent relates to "communication networks, and ... distributed bridging systems." Ex. 1001 at 1:6–9. "Local Area Networks (LANs) connect computing systems together at the Layer 2 level. The term "Layer 2" refers to the second layer in the protocol stack defined by the well-known Open Systems Interface (OSI) model, also known as the logical link, data link, or Media Access Control (MAC) layer." *Id.* at 1:13–17. The '400 patent incorporates by reference the IEEE 802.1D standard and describes that "MAC bridges that implement the 802.1D standard allow MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* at 1:27–30. The specification also describes that "Recently, various means have been proposed and developed for transporting Layer-2 packets, such as Ethernet frames, over high-speed, high-performance Layer-3 packet networks." *Id.* at 1:47–51.

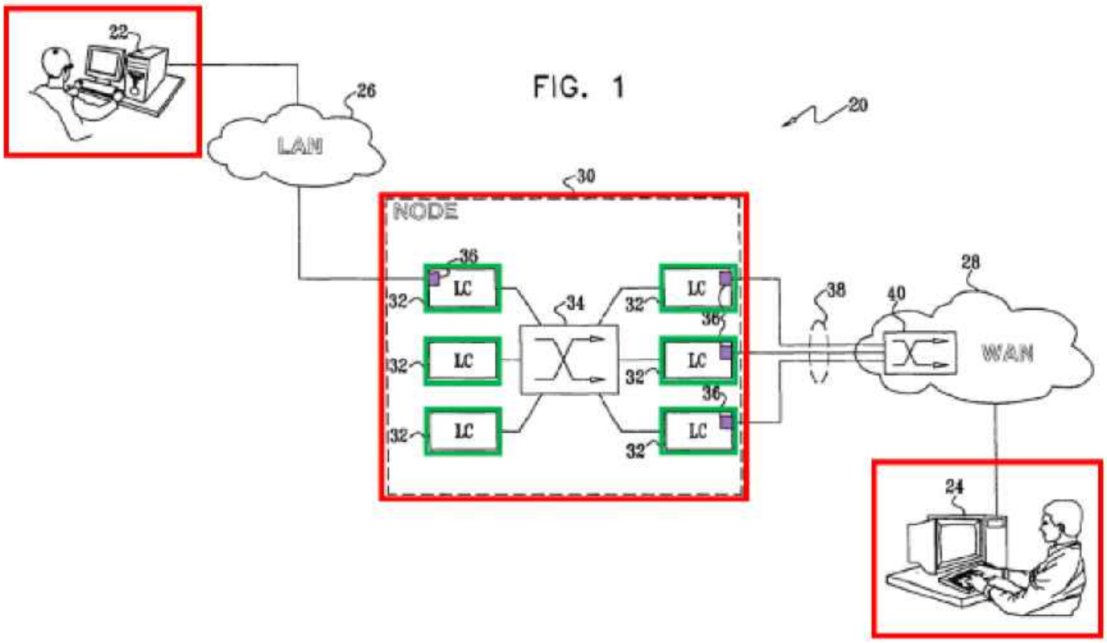
63. At the time of the invention, "a number of authors have described methods for creating a virtual private LAN service (VPLS), which links different

LANs together over an IP network." *Id.* at 2:9–12. Further, the specification describes that "Link aggregation (LAG) is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link (referred to as the "LAG group")." *Id.* at 2:37–40. "Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic." *Id.* at 2:40–42. The '400 patent states that "For ethernet networks, link aggregation is defined by Clause 43 of IEEE Standard 802.3, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications (2002 Edition), which is incorporated herein by reference." *Id.* at 2:43–47.

64. Figure 1 of the '400 patent (reproduced below) shows that "terminal **22** is connected to a LAN, such as an Ethernet LAN, while terminal **24** is connected to a wide area network (WAN) **28,** such as the Internet or another Layer 3 network. The VPLS, however, permits the users of terminals **22** and **24** to communicate with one another as though they were connected to the same LAN domain." *Id.* at 5:57–62. Further, the '400 patent states, "The specific configuration of LAN **26** and WAN **28** is shown in FIG. **1** purely by way of illustration, and the principles of the present invention may be applied in substantially any network configuration that supports the provisioning of Layer 2 virtual private networks." *Id.* at 6:2–7. The '400 patent further describes Figure 1 as follows:

> "In the exemplary configuration shown in FIG. 1, a network node 30 links LAN 26 and WAN 28. Node 30 comprises multiple line cards 32, linked by a switching core 34. Line cards 32 have ports 36, which connect to other nodes in LAN 26 and WAN 28 (and possibly in other networks, as well). Typically, each line card comprises multiple ports, although only a few ports are shown in FIG. 1. In the description that follows, ports 36 are assumed to be Ethernet ports, for the sake of simplicity of explanation. Alternatively, some or all of the line cards may comprise ports of other types, and may connect to other types of networks, such as Internet Protocol (IP) networks. For example, in an alternative embodiment (not shown in the figures), WAN 28 comprises a Resilient Packet Ring (RPR) network, and some of line cards 32 thus comprise RPR interfaces. Features of a network node that may be used to connect an Ethernet network to a RPR network are described, for example, in U.S. patent application Ser. No. 10/993,882, filed Nov. 19, 2004, which is assigned to the assignee of the present patent application and whose disclosure is incorporated herein by reference. Additionally, or alternatively, line cards 32 may connect to tunnels, such as Multi-Protocol Label-Switching (MPLS) tunnels, through WAN 28 via appropriate label-switched routers in the WAN."

Id. at 6:8–30.



65. As stated above, a network node 30 "comprises multiple line cards 32, linked by a switching core 34," and each line card has ports 36, which connect to other nodes in LAN 26 and WAN 28. *Id.* at 6:8–12. "MAC bridges maintain a forwarding database (FDB) to map destination MAC addresses of the packets they receive to bridge ports. The bridge builds the forwarding database by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received. When the bridge receives an incoming packet whose destination address is not found in the database, it floods (*i.e.*, broadcasts) the packet through all its available ports, except the one through which the packet arrived. Other MAC bridges that do not recognize the destination address will further flood the packet to all the relevant ports. Through the flooding mechanism, the packet will eventually traverse all interconnected bridges at least

once, and will ultimately reach its destination." *Id.* at 1:33–46. The '400 patent states:

> "Each of the line cards may typically serve as both ingress and egress for data packets and has a respective MAC forwarding database (FDB) that is shared by the ingress and egress functions. When an ingress line card receives an incoming data packet over the VPN on one of its ports, it consults the FDB in order to choose the line card and port through which the packet should be forwarded based on the MAC destination address (or floods the packet through the ports in the VPN when the MAC destination address does not appear in the FDB)."

Id. at 3:7–17. As I discussed above, this method of flooding was well known at the time of the '400 patent. The '400 patent allegedly identified a need for better MAC address learning at the bridges within a VLAN to mitigate the need to flood (or broadcast) messages when the destination MAC address was unknown, as follows:

> "When the forwarding destination of a packet is a link aggregation group (LAG), LAG member selection (i.e., selection of the link over which the packet is to be forwarded) is typically performed on the ingress line card. In the absence of the synchronization method described above, other members in the LAG may not receive such packets for transmission, so that the FDB of the corresponding line cards will not be updated. When these line cards receive incoming packets, the result may be constant flooding, since the FDB is incomplete. The synchronization mechanism described herein avoids this

> problem by **updating the FDB in all line cards in the LAG group (or across the entire VPN instance**) within the node. Typically, when the transmitting line card transmits the data packet via a port that belongs to a LAG group, the synchronization message sent by the line card identifies the VPN instance and the incoming port. The **other line cards in the same LAG group (as well as all the other line cards serving this VPN instance**) can use this information to learn the MAC address association **even when these other line cards have not yet received packets from the MAC address in question**."

Id. at 3:34–53 (emphasis added).

66. Referring to Figure 2, the '400 patent describes:

> Upon receiving an incoming packet from switch **40,** port **36** passes the packet to ingress path **54**. Packet processor **52** identifies the VPLS (typically by a lookup and classification process based on certain packet header fields), extracts the other key parameters from the incoming packet (including the MAC destination address (DA), and optionally, the VLAN 30 identifier), and uses the key to query database ***58***. If the record is found, the packet processor adds a tag to the packet indicating the egress port through which the packet should be forwarded, as well as the ingress port through which the packet was received. If the output interface indicated by the record is a LAG group, the packet processor selects one of the physical ports in the LAG group (using a hash function, for example), and tags the packet for transmission via the selected port. The packet processor then

> passes the tagged packet to switching core ***34***, which conveys the packet to egress path ***56*** of the appropriate port.
>
> When packet processor ***52*** receives a packet on ingress path ***54*** for whose key there is no a corresponding record in database ***58***, however, it tags the packet for flooding. In this case, switching core ***34*** will pass the packet for transmission via all the ports (other than the ingress port through which the packet was received) that are used by this VPLS instance. For each LAG group serving the VPLS instance, however, the flooded packet is transmitted via only one port in the group.

Id. at 7:25–49. That is, upon receiving an incoming packet, "[i]f the record is found, the packet processor adds a tag to the packet indicating the egress port through which the packet should be forwarded." *Id.* at 7:31–34. If there is "no . . . corresponding record in the database . . . however, it tags the packet for flooding." *Id.* at 7:42–44. While the "switching core 34 will pass the packet for transmission via all ports . . . that are used by this VPLS instance," "[f]or each LAG group serving the VPLS instance, however, the flooded packet is transmitted via only one port in the group." *Id.* at 7:47–49.

67. The patent explains that if the packet is transmitted using a LAG, a single port within the LAG is chosen and transmitted only through that port, including when the packet is flooded. *Id.* at 6:31-48,7:47-49. However, given that the other members of the LAG may not receive such packets, their FDBs may not

be updated, which may result in unnecessary flooding when dealing with future packets. *Id.* at 3:34-53.

68. To address this, the purported invention provided "improved methods for MAC learning" that are "useful especially in the context of nodes that are configured to serve as virtual bridges in Layer 2 virtual private networks … particularly when multiple ports of the node are conjoined in a LAG group." *Id.* at 2:60-3:2. The patent discloses a known technique of MAC source learning on the "egress path." *Id.* at 3:18-24, FIG. 3, 7:55-63. The patent explains that "[l]earning on egress is advantageous particularly with respect to flooded packets, since in this case multiple line cards receive the packet and are able to learn the interface association of the MAC source address (SA) and VPLS instance." *Id.* at 7:55-63.

69. After a line card learns the MAC source address and VPLS instance on the egress path, the line card provides a synchronization message to other line cards to update their FDBs with the learned information. *Id.* at 8:17-29,9:47-65. Synchronization messages ("SYNC") are sent at regular intervals "to report each SELF entry that it has created in the FDB 58 to the other line cards 32 in node 30." *Id.* at 8:17–22. The MAC table differentiates between SELF entries, which are entries that are learned by the packet processor on the line card and SYNC entries. *Id.* The '400 patent uses an aging mechanism that is applied to the "MAC database 58 in order to remove records that are no longer in effect and free space for new

records." *Id.* at 9:4–6. A record is removed from the database "if a predetermined aging time elapses following the timestamp without a further packet having been received with the same key." *Id.* at 9:9–11. If, on the other hand, the current packet matches a record in the FDB, "the packet processor refreshes the timestamp of the record," and "forwards the packet to the appropriate output port." *Id.* at 9:29–31.

B. File History of the '400 Patent

70. As I state below, in forming my opinions, I have also reviewed the file history of the '400 patent.

71. The '400 patent's application was filed on May 19, 2006. Ex. 1001, [22]. The Examiner rejected certain pending claims as anticipated by Edsall, or obvious over Edsall in view of another reference. Ex. 1002, 81–85. During an interview between the applicants and the Examiner, it was discussed that if the limitation "link aggregation group having a plurality of ports" in claim 1 were amended to recite the claim term as defined by the specification (at 2:37–40), then the amended claim would overcome Edsall. *Id.* at 112. The applicant therefore amended the claim limitation to add "a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link." *Id.* at 114. The applicant made no other amendments to overcome the rejections over Edsall. The Examiner allowed the claims, and the patent issued on September 22, 2009. *Id.* at 128, 160.

72. Based on my review of the record, I understand that the claims were amended to add "a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link" to overcome the rejection by Edsall, which corresponds to the language found in claims 1[b] and 11[c]. My opinions do not rely on Edsall for disclosure of this particular claim language that was added to claims 1[b] and 11[c]. Instead, it is my opinion that claim 1[b] and 11[c] are disclosed by Smith, which I describe in more detail below.

VII. MATERIALS AND PRIOR ART CONSIDERED

73. I have considered information from various sources in forming my opinions. I have drawn on my decades of experience in this field. I have employed methods and analyses of a type reasonably relied upon by experts in my field in forming opinions or inferences on the subject. Additionally, in preparing this Declaration, I have relied upon '400 patent and its file history, the exhibits cited to below, and the additional exhibits listed at the beginning of this Declaration.

74. In this Declaration, I provide six prior art references with exemplary citations identifying the relevant features related to the claim language of the Challenged Claims.[2] I have also described combinations of these prior art references

[2] The complete text of the Challenged Claims is set forth in Appendix B hereto.

that, when combined, render the Challenged Claims of the ’400 patent obvious. It is my opinion that the Challenged Claims of the ’400 patent are rendered obvious in light of the prior art specifically discussed in this Declaration.

75. The citations presented in the body of this declaration correspond to the teachings disclosed in exemplary items of prior art that identify: (1) the problem(s) confronted by one of ordinary skill in the art; (2) an express suggestion to make one or more of the combinations; (3) an implicit suggestion to make one or more of the combinations; or (4) the knowledge of those skilled in the art as to the applicable field of technology at the relevant timeframe.

76. It is my opinion that a person of ordinary skill would consider at least the prior art references below:

Prior Art Reference	Date of Publication	Exhibit No.
U.S. Patent App. Pub. No. 2005/0198371 (“Smith”)	September 8, 2005	Ex. 1004
U.S. Patent No. 6,999,418 (“Sharma”)	June 26, 2003	Ex. 1026
Japanese Patent App. No. 2005086668 (“Ishimori”)	March 31, 2005	Ex. 1005
U.S. Patent No. 6,735,198 (“Edsall”)	May 11, 2004	Ex. 1006
U.S. Patent App. Pub. No. 2004/0133619 (“Zelig”)	July 8, 2004	Ex. 1007
IEEE 802.1Q-1998 (“802.1Q-1998”)	March 8, 1999	Ex. 1008

77. Specifically, it is my opinion that the claims of the ’400 patent would have been obvious as follows:

’400 Patent Claims	Basis of Invalidity
1, 3, 6, 11, 13, 16	35 U.S.C. § 103 over Smith and Sharma
1–3, 11–13	35 U.S.C. § 103 over Smith, Sharma, and Ishimori
1, 4–7, 10, 11, 14–17, 20	35 U.S.C. § 103 over Smith, Sharma, Ishimori, and Edsall
8–9, 18–19	35 U.S.C. § 103 over Smith, Sharma, Ishimori, and Zelig
9, 19	35 U.S.C. § 103 over Smith, Sharma, Ishimori, Zelig, and 802.1Q

78. I am also not aware of any secondary considerations of non-obviousness that affect my conclusions regarding the obviousness of these claims.

79. I understand that defendants in a district court case filed by Corrigent Corporation, Cisco Systems Inc., Dell Technologies Inc., and Dell Inc. (“Cisco”), have also filed an IPR challenging the validity of the ’400 Patent. I have reviewed the petition filed by Cisco referred to here as Ex. 1028. I have also reviewed the declaration by Cisco’s expert, Dr. Bambos, attached as Ex. 1029. I have never met with or discussed the validity of the ’400 Patent with Dr. Bambos. I agree with and incorporate by reference the totality of Dr. Bambos’s declaration with respect to his interpretation of the prior art references. However, I have a slightly different interpretation of the claims of the ’400 patent. Thus, my application of the prior art

references differs slightly. I have also added an additional reference, Sharma, to support invalidity grounds raised in my declaration.

80. Other references and knowledge described earlier in this declaration are relevant to show the state of the art (*i.e.*, what a person of ordinary skill would have known when using the references identified above) or to use in combination with the references above to render the Challenged Claims invalid. I have considered also the following, which are also either cited or incorporated by reference in the '400 patent:

a. IEEE 802.1D, Media Access Control (MAC) Bridges, 2004 Edition (Ex. 1015)

b. IEEE Standard 802.1Q, Virtual Bridged Local Area Networks, 2005 Edition (Ex. 1016)

c. IEEE 802.3 Standard for Local and metropolitan area networks: Specific requirements, 2002 Edition (Ex. 1017)

d. Kompella et al., *Virtual Private LAN Service*, IETF (December 2005) (Ex. 1018)

e. Lasserre et al., *Virtual Private LAN Services over MPLS*, IETF (November 2005) (Ex. 1019)

f. Martini et al., *Encapsulation Methods for Transport of Ethernet Over MPLS Networks*, IETF (November 2005) (Ex. 1020)

g. U.S. Patent No. 6,917,986 (Ex. 1021)

h. U.S. Patent No. 7,974,223 (Ex. 1022)

VIII. INVALIDITY ANALYSIS OF THE CHALLENGED CLAIMS

81. Below is a detailed analysis of the bases of invalidity identified above.

82. I note that claim 1 in the '400 patent is a method claim and claims 2 through 10 depend from claim 1; claim 11 is an apparatus claim and claims 12 through 20 depend from claim 11. I am informed by Counsel that a method claim is a claim that describes a series of acts or steps for performing a desired function or accomplishing an intended result. I am informed by Counsel that an apparatus claim is a claim that defines the invention by its physical or structural components and their relationships. Because of the two claim types, there exist slight differences in the claim language (*e.g.*, "sending" versus "send"). Additionally, the apparatus claim 11 recites a switching core, which is not found in independent claim 1. However, besides these differences, the differences found in the other claim limitations are immaterial for the purposes of my conclusions regarding unpatentability and do not affect my analysis. For my analysis, I have grouped claim limitations where they are substantively similar or where it made sense to discuss them together.

A. Overview of Prior Art References

1. Smith

83. Smith is directed to a virtual network device comprising interface bundles, which are managed as a single logical interface. Smith discloses:

> "A virtual network device includes several different virtual network device sub-units, which collectively operate as a single logical network device. An interface bundle includes interfaces in more than one of the different virtual network device sub-units included in the virtual network device. The interface bundle is coupled to a virtual link bundle, which connects the virtual network device to another device. The interface bundle is managed as a single logical interface."

Ex. 1004, Abstract; *see also id.* ¶ 34. Smith teaches that "[t]he communication links are configured to be managed as a single link. When the first network device sends a packet to the virtual network device via the virtual link bundle, the first network device selects one of the communication links on which to send the packet. Each packet sent between the virtual network device and the first network device is sent via only a one of the communication links." *Id.* ¶ 9.

84. As depicted in Figure 3, virtual network device 202 is coupled to other network devices 120(1)–120(3). *Id.* ¶ 44. The virtual network device consists of virtual network device sub-units 122(1) and 122(2), which includes several line cards 304(1)–304(4). *Id.* ¶ 46. "In virtual network device sub-unit 122(1), line card 304(1) includes forwarding engine 314(1) and interfaces 320(5), 320(7), and 320(9). Interface 320(7) is coupled to network device 120(3). Interface 320(9) is also

coupled to network device 120(1)." *Id.* ¶¶ 46–47. "Line card 304(4) includes forwarding engine 314(4) and interfaces 320(12), 320(14), and 320(16). Interfaces 320(12) and 320(16) are respectively coupled to satellite network devices 120(3) and 120(1)." *Id.* ¶ 48. "Interfaces 320(13), 320(9), and 320(16), which are each coupled to network device 120(1) by virtual link bundle 250(1), form an interface bundle (e.g., an EtherChannel (TM) port bundle)." *Id.* ¶ 53. "Similarly, interfaces 320(11) and 320(8) form another interface bundle that is coupled to network device 120(2) by virtual link bundle 250(2). Interfaces 320(7) and 320(12) form a third interface bundle that is coupled to network device 120(3) by virtual link bundle 250(3)." *Id.*

85. The interface bundle is described by Smith as follows:

> "One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using **a link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**. For example, an EtherChannel (TM) port bundle can be formed from several ports on a switch, each of which is coupled to a respective link in a group of links coupling that switch to another switch. Once an EtherChannel (TM) port bundle is formed, the port bundle can be managed as a single bridge port by routing protocols such as spanning tree, thus simplifying management of the redundant links."

Id. ¶ 6 (emphasis added).

> "Various embodiments of methods and systems for implementing interface bundles in virtual network devices are disclosed. A virtual network device includes several different virtual network device subunits, which collectively operate as a single logical network device. An interface bundle includes interfaces in more than one of the different virtual network device sub-units included in the virtual network device."

Id. ¶ 8.

86. Additionally, Smith teaches that:

> "The redundant links coupling each of network devices 120(1) and 120(2) to virtual network device 202 can be operated as a **single logical link, referred to herein as a virtual link bundle**. Network device 120(1) operates the two links coupling network device 120(1) to virtual network device 202 as a virtual link bundle 250(1). In such an embodiment, each interface in network device 120(1) that is coupled to one of the links is included in an interface bundle, which corresponds to virtual link bundle 250(1). Network device 120(2) similarly operates the two links coupling network device 120(2) to virtual network device 202 as virtual link bundle 250(2). In some embodiments, **virtual link bundles** 250(1) and 250(2) are each operated as an EtherChannel (TM) or **as an aggregated link (as described in IEEE 802.3)**."

Id. ¶ 36 (emphasis added).

87. When a packet is received at an uplink interface, the virtual network device sub-unit can "learn that the sending device's MAC (Media Access Control)

address is "behind" uplink interface 320(13) by associating the MAC address with the logical identifier of uplink interface 320(13)." *Id.* ¶ 54. The sub-unit then informs each forwarding engine within the virtual device of this association (between the MAC address and the logical identifier of the uplink interface). *Id.* As Smith teaches, "packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier." *Id.*

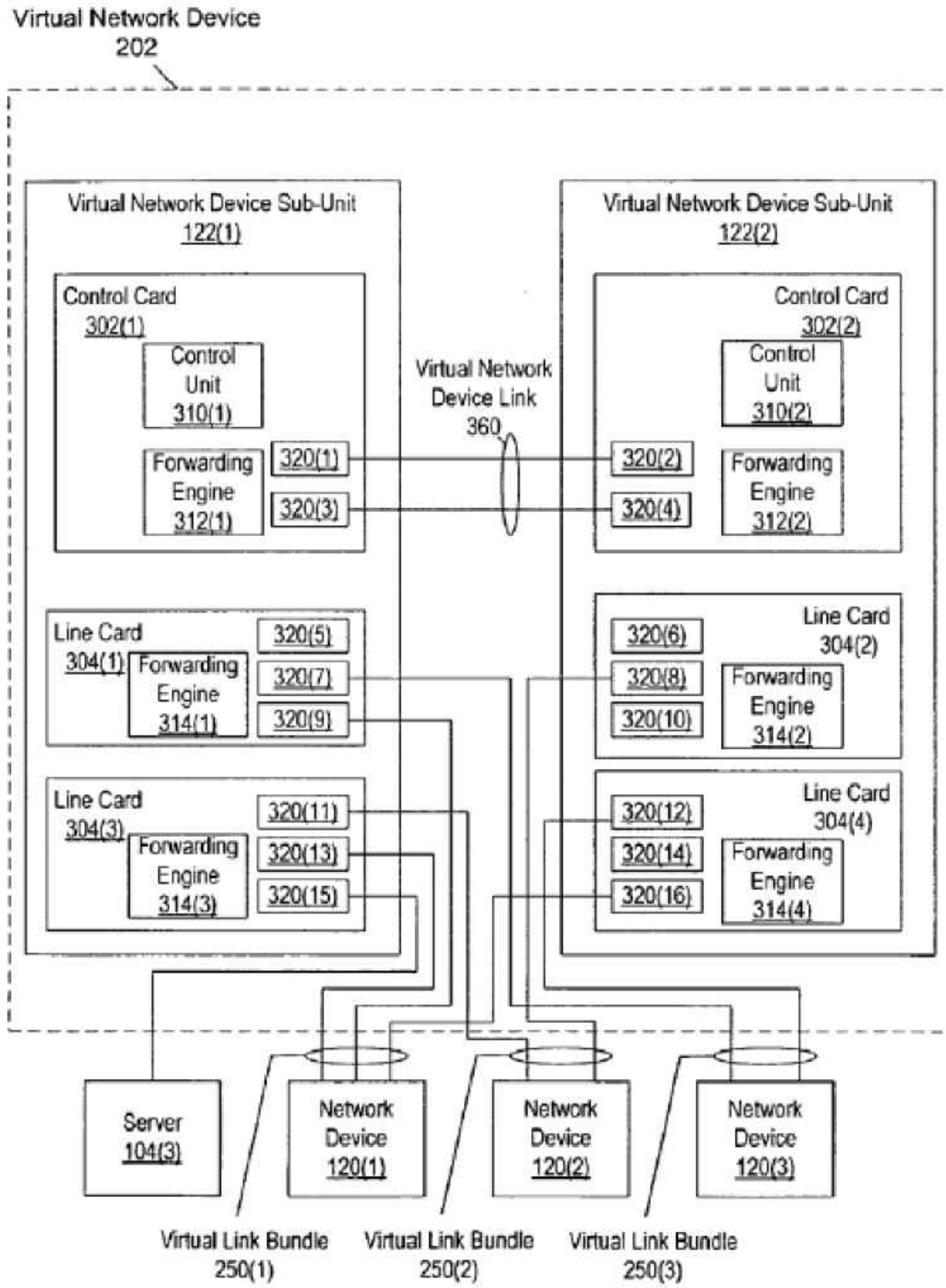


FIG. 3

88. In order to keep its MAC tables updated, Smith uses "MAC notification frames are used to keep the content of the L2 tables in virtual network device subunit 122(1) synchronized with the content of the L2 tables in virtual network device sub-

unit 122(2) and vice versa." *Id.* ¶ 62. If a "forwarding table already includes an entry associating the destination address with a port of one of the network devices," that forwarding engine will generate "a MAC notification identifying this association, which is distributed to any other forwarding engines within virtual network device sub-unit 122(2)." *Id.* ¶ 63. "If there is no hit in the forwarding table, as determined at 407, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at 409." *Id.* ¶ 66. But to take advantage of the virtual link bundles, Smith further specifies that as a result of the virtual link bundling, the data packet is "sent via only [] one of the communication links." *Id.* ¶ 9.

2. Sharma

89. Sharma is directed to a technique for "reduc[ing] the incidence of unnecessary frame flooding," particularly "in connection with aggregated ports." EX1026 2:11-18. Sharma's framework utilizes a known learning process referred to as "egress learning," where a frame's source address is learned as the frame exits

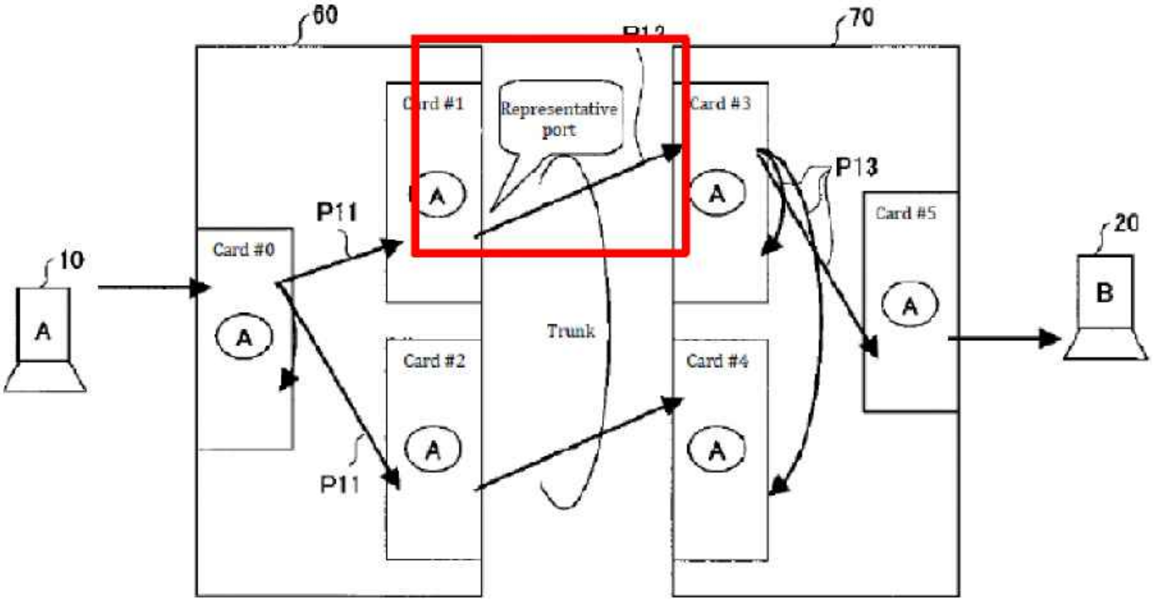
a network device through an egress port, as opposed to learning at an ingress port. *Id.* 1:60-2:7,1:41-51,4:6-11,3:3-4.

90. An aggregated port ("AP") is a set of physical ports that are logically grouped together such that the grouped ports appear as a single logical connection. *Id.* 1:52-59. According to Sharma, when employing an AP, a particular port may be selected for transmitting frames in one direction while a different port may be selected for transmitting frames in the other direction. *Id.* 4:33-5:10. In such a scenario, address information learned by one port of the AP may not be available to another, and thus frames sent through the AP may be continuously flooded "in a wasteful manner, despite the existence of information at another port of the device that could be used to terminate the flooding." *Id.* 1:60-2:7. One known solution to this issue is to periodically synchronize the forwarding tables of all ports, such that the information learned by one port is shared with other ports. *Id.* 5:11-21. However, Sharma states that "such a mechanism is resource-intensive," and "[r]eliance upon this mechanism alone may result in relatively inefficient operation." *Id.* 5:21-25. The purported invention of Sharma provides another solution, where the forwarding tables are selectively synchronized by counting the number of instances a particular port floods a frame with an unknown destination address, and if that number reaches a predetermined threshold, inferring that another port has the

necessary information about that destination address, and sharing that information with the particular port. *Id.* 2:19-42.

3. Ishimori

91. Ishimori is directed to a packet forwarding method to "prevent repeated flooding" in systems using "path learning" and "link aggregation." Ex. 1005, Abstract. Ishimori discloses a system with node groups 60, 70 in between two terminals A10 and B20. *See* Ex. 1005, Fig. 9. The node groups 60, 70 have communication cards #0, #1, #2 and #3, #4, #5, respectively. *Id.* ¶ 15. Ishimori teaches a method where the source MAC address of a received packet is learned by storing the source MAC address "in a buffer [MAC table] had by each node." *Id.* ¶ 2. Thus, the destination MAC address for a packet is searched for "in the MAC table of the local device." *Id.* ¶ 4. If the result is found in the buffer (MAC table), then the packet is transmitted according to the forwarding path that was learned. *Id.* Thus, Ishimori teaches that "as long as a learning result relating to the destination MAC address (DA) of the receive packet exists in the buffer, there is no need for flooding and traffic in the connected LAN can be kept to a minimum, since the forwarding path to apply to the corresponding packet can be decided unambiguously by using this information." *Id.* ¶ 6.



[FIG. 9] Block diagram for describing one example of conventional link aggregation function

92. However, if the MAC address is not found, Ishimori acknowledges that in previous approaches, the packet would be flooded to all nodes. *Id.* ¶ 2. Ishimori recognized, however, the inefficiency in this approach because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶ 19. Ishimori then teaches a solution to this flooding problem. Specifically, Ishimori teaches, by leveraging "link aggregation," which bundles a plurality of ports to function as one virtual port, that "one representative port is selected from among the large number of ports." *Id.* ¶ 13. And the packet is therefore transmitted "using this representative port." *Id.* Thus, "a packet received

from the terminal 10 and addressed to terminal 20 is received at card #0" of node 60. *Id.* ¶ 15. "One representative port is selected . . . and this packet is further transmitted." *Id.* Ishimori also teaches that for "each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16. Ishimori also teaches that a "learn packet" is generated at a "predetermined timing." *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33.

93. Ishimori also teaches an aging process. When the buffers of the cards of each node learn a MAC address, the "hit bit" corresponding to the learned MAC address is set to "1." *Id.* ¶ 9. Then, after the first aging cycle, the hit bit is set to "0," which indicates that the MAC address should be deleted from the buffer on the next cycle. If the MAC address is learned again at the next aging process, the hit bit is again set to "1." *Id.*

4. Edsall

94. Edsall is generally directed to techniques for updating and synchronizing "forwarding tables contained on line cards that are interconnected by a switch fabric of a distributed network switch." Ex. 1006 at Abstract. Edsall teaches that the "forwarding table" has an L2 portion that is "used to execute forwarding

decision operations for frames forwarded among ports of the line cards." *Id.* at 5:66–6:4. Similar to Smith and Ishimori, in Edsall, "[i]f the frame is received at the ingress card for the first time, this ingress forwarding engine also 'learns' a source MAC address of the frame." *Id.* at 6:26–31. This involves "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 6:31–34. "The ingress forwarding engine then performs a flood-to-fabric (FF) operation on the frame by asserting all bits in the POE mask field of the fabric frame. The asserted POE bits instruct the switch fabric to switch ("flood") copies of the fabric frame through its port interfaces to all (egress) line cards of the network switch." *Id.* at 6:34–39; *see also id.* at 18:44–47.

95. Edsall further teaches a "novel MN frame is provided to complement the FF operation." *Id.* at 6:46–50. The MN (MAC notification) frame "involves use of a primary input (PI) indicator," which "denotes a primary input MAC address that is directly attached to a port of the line card associated with the forwarding table containing this entry." *Id.* at 6:50–56. The "PI indicator is asserted for a forwarding table entry having a MAC address that is learned from a frame sourced through one of the ports of the line card, as opposed to being learned through the switch fabric." *Id.* at 6:56–60; *see also id.* at 18:56–19:5. The frame additionally includes a POE (port-of-exit) field that "includes a plurality of bits, one for each port interface of the switch fabric." *Id.* at 6:24–25. The POE bit instructs the switch which port interfaces

on which line cards should receive the MN frame. *Id.* at 9:47–50. Edsall also teaches an aging process "directed to aging of entries in the forwarding tables of the distributed switch." *Id.* at 17:41–42. "[A] MAC address entry that has not been refreshed as a source within a specified period of time is removed from the L2 portion of the forwarding table in connection with a conventional aging policy executed by the microprocessor on the line card." *Id.* at 17:43–47; *see also id.* at 17:41–18:34.

5. Zelig

96. Zelig is directed to a data communication network that "includes a plurality of primary virtual bridges, interconnected by primary virtual connections." Ex. 1007, Abstract. Zelig discloses that "MAC bridges that implement the 802.1D standard allow MAC devices attached to physically separated LANs to appear to each other as if they were attached to a single LAN." *Id.* ¶ 3. A MAC bridge "includes two or more MAC devices that interconnect the bridge ports to respective LANs." *Id.* The "MAC bridges maintain a database to map destination MAC address of the packets they receive to bridge ports." *Id.* ¶ 42. "The bridge builds the database by means of a learning process, in which it associates the source MAC address of each incoming packet with the port on which the packet was received." *Id.* Zelig provides "improved mechanisms for protection from failure in virtual private networks (VPNs)." *Id.* ¶ 18. Zelig thus discloses a data communication network

including "a plurality of primary virtual bridges" and "one or more backup virtual bridges" that are "arranged to transmit the packets using a virtual private LAN service (VPLS)." *Id.* ¶¶ 21–24. "[E]each of the primary and backup virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table." *Id.* ¶ 27. The VPN 20 includes multiple primary virtual bridges. *Id.* ¶ 42.

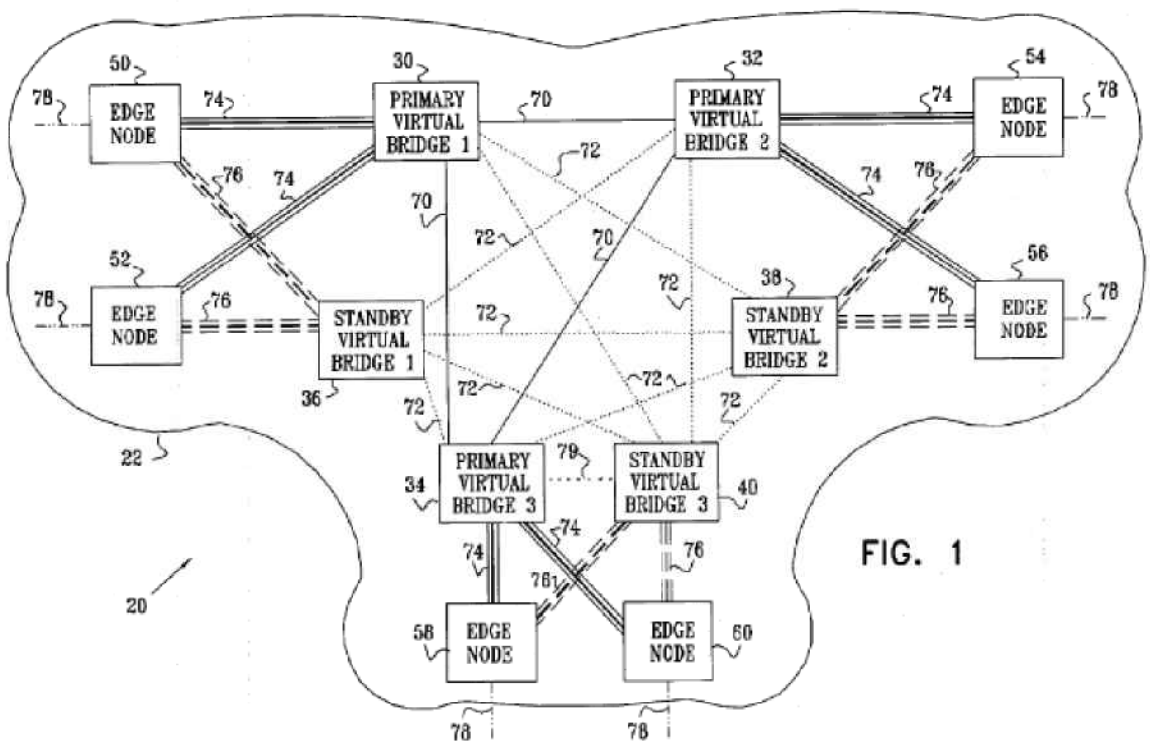


FIG. 1

6. 802.1Q-1998

97. The 802.1Q-1998 standard defines the architecture for Virtual Bridged LANs, its services, protocols, and algorithms. Ex. 1008, Abstract. It was part of the effort of the IEEE to standardize virtual LAN services in bridged LANs. The standard discloses a twelve-bit VLAN identifier field that is used to "uniquely

identify the VLAN to which the frame belongs." *Id.* § 9.3.2.3. This standard would have been known to a POSITA at the time of the '400 patent.

B. Motivations to Combine

1. Smith and Sharma

98. A POSITA considering the teachings of Smith would have also considered the teachings of Sharma, as they are analogous prior art pertaining to the same field of endeavor, namely, forwarding data packets through a communication network employing aggregated, or bundled, ports and links. *Compare* EX1004 ¶¶2,6 *with* EX1026 1:16-19,1:52-59. A POSITA would have been motivated to combine the teachings of Smith and Sharma for multiple reasons, as discussed below.

99. **First**, the framework Sharma's invention is substantially identical to that of Smith. Smith discloses techniques for forwarding a data packet through a network device employing aggregated links. EX1004 ¶¶6,9,30,50. Sharma similarly discloses techniques for forwarding a data packet through a network device employing aggregated ports. EX1026 1:20-28,1:52-59,4:33-5:25. The only difference between the two disclosures is the use of alternatives source learning processes. Sharma's framework utilizes a known learning process referred to as "egress learning," where a frame's source address is learned as the frame exits a network device through an egress port. *Id.* 1:60-2:7,1:41-51,4:6-11,3:3-4. In contrast, Smith's framework learns a frame's source address when the frame is first

received by a network device (referred herein as "ingress learning"). EX1004 ¶0065. Egress learning was a known substitute of ingress learning. There is nothing uniquely challenging or difficult about modifying a network device to learn a frame's source address when the frame is first received by the network device as opposed to when the frame is to be transmitted out of the network device. A POSITA would have been motivated to combine the teachings of Smith and Sharma based on simple substitution and would have had a reasonable expectation of success doing so.

100. **Second**, it would have also been obvious to combine the teachings of Sharma and Smith because Sharma expressly identifies a technique used by Smith and provides improvements to it. Sharma explains that the traditional methods for synchronizing the forwarding tables are inefficient and wasteful, particularly because flooding itself is resource-intensive. EX1026 5:16-33. Smith's framework utilizes these traditional methods. EX1004 ¶63. Sharma then provides an improved way of synchronizing the forwarding tables by selectively updating them by counting the number of instances a particular port floods a frame with an unknown destination address, and if that number reaches a predetermined threshold, inferring that another port has the necessary information about that unknown destination address, and instructing that port to share that information. EX1026 2:19-42. A POSITA looking to improve the efficiency of operating a network device would

have been recognized the problem that Sharma described, recognized that Smith's system also shared that problem, and thereafter have been motivated to implement Sharma's technique of selectively updating the forwarding tables to Smith's system.

2. Smith, Sharma, and Ishimori

101. A POSITA would have been motivated to combine the teachings of Smith, Sharma, and Ishimori, because all references disclose link bundling and path learning techniques in communication networks and sending packets through one port of the link bundle. As explained above, Smith discloses that "[e]ach packet sent between the virtual network device and the first network device is sent via only a one of the communication links." Ex. 1004 ¶ 9. Sharma discloses the same. EX1026 1:57-59. Smith further teaches that "[f]or interfaces (e.g., ports or uplinks) included in interface bundles, the virtual network device subunit selects one egress interface per interface bundle via which to send the packet." Ex. 1004 ¶ 66.

102. Similarly, Ishimori also discloses flooding via one representative port. Ex. 1005 ¶ 13. Ishimori recognized the shortcomings in the approach taken by others of flooding all ports because the learning results on the communication cards in the same trunk are not leveraged and "flooding is constantly performed." *Id.* ¶ 19. Ishimori explains that using "one representative port is selected from among the large number of ports had by this trunk according to a predetermined computation, and actual packet transmission is performed using this representative port." *Id.* ¶ 13.

"This configuration prevents the application of a different path in the same trunk to the same packet." *Id.* ¶ 14.

103. A POSITA looking to solve the shortcomings of flooding all the ports would have therefore looked to Smith, Sharma, and Ishimori. Smith teaches that selecting one egress interface on a virtual network bundle. Ex. 1004 ¶ 66. Ishimori teaches that using "one representative port" prevents constant flooding. A POSITA looking to implement this solution would have therefore been motivated to combine their teachings because flooding over only one link in a bundle would prevent sending similar packets unnecessarily over multiple links. A POSITA would have understood that flooding would also not occur over the ingress port.

104. Additionally, Ishimori teaches a "learn packet" that is generated at a "predetermined timing." Ex. 1005 ¶ 25. If one of skill in the art were looking to limit flooding over unnecessary links, one would have been motivated to implement Ishimori's learn packet generated at a predetermined timing into Smith's MAC notification method. Furthermore, one of skill in the art would have had a reasonable expectation of success in implementing this combination because it would have required a simple substitution of Ishimori's structures for Smith's relevant structures. Both Smith and Ishimori leverage link bundling/aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within

the knowledge of a POSITA implementing such a method to combine the teachings of Smith and Ishimori and have a reasonable expectation of success.

3. Smith, Sharma, Ishimori, and Edsall

105. A POSITA would have been motivated to combine Smith, Sharma, Ishimori, and Edsall. The Smith-Sharma-Ishimori combination would have been a virtual network device with several virtual network device sub-unit operating as a single logical network device with MAC bridges. Ex. 1004 ¶ 54; Ex. 1026 3:42-50; Ex. 1005 ¶ 9; Ex. 1006 at Abstract. It would have been obvious to a POSITA to modify the MAC forwarding tables, as disclosed in Smith, Sharma, or Ishimori, to implement the various features found in Edsall's forwarding engine. Specifically, a POSITA could implement the MAC tables to implement how Edsall's forwarding engine learns the source MAC address by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." Ex. 1006 at 18:39–44. A POSITA could further implement Edsall's "flood-to-fabric (FF) operations" by implementing the MAC tables to include a POE bits that determines which port interfaces on which line cards should receive the MN frames. *Id.* at 18:47–50. Finally, it would have been obvious to a POSITA to implement the MAC tables to include the PI indicator as taught in Edsall in order to keep track of whether a MAC address was learned from a local line card or a different line card. It would have also been obvious to a POSITA to further

implement the MAC forwarding tables of Smith-Sharma-Ishimori to use the POE field in order to determine which ports should receive the MN frame.

106. Furthermore, creating entries in a MAC forwarding table and adding a PI indicator field and POE field to the MAC tables would have been an implementation that a POSITA would have known how to make with a reasonable expectation of success.

107. Ishimori additionally teaches an aging process. Ishimori's aging process uses a "hit bit." Ex. 1005 ¶ 9. When a MAC address is first learned, the hit bit is set to 1. *Id.* Then, "at the first aging process, this hit bit is cleared," meaning the "hit bit" it is set to 0. *Id.* The learning result is not yet deleted. *Id.* If on the next aging cycle, the MAC address is learned again, the hit bit will be set again to 1. *Id.* Otherwise, the entry will be deleted. *Id.* Therefore, the "hit bit" keeps track of how long a MAC address has been in the MAC address table without having been refreshed. *Id.*

108. A POSITA would have understood that MAC addresses may become stale due to changes in the network and would have been motivated to look at solutions for making sure the MAC table is up to date. Although Smith does not disclose aging the entries in the MAC table, it would have been obvious to a POSITA to look for a type of aging parameter could have been used and Ishimori describes precisely that. Moreover, because the size of a MAC table is limited, a POSITA

would have known that older entries have to be deleted when excessively aged to make space for newer entries. A POSITA looking to implement this solution would have therefore been motivated to combine the teachings of Smith and Ishimori and it would have been an easy application of Ishimori's methods to Smith's system. For example, the aging process could be added to the MAC address table described in Smith using known techniques and a simple addition of that information to a table. *Id.*

109. Furthermore, one of skill in the art would have had a reasonable expectation of success in implementing this combination because it would have required a simple implementation of Ishimori's aging process to Smith-Sharma-Edsall. Smith, Sharma, Ishimori, and Edsall already leverage link aggregation techniques in the packet forwarding methods they disclose. Thus, it would have been within the knowledge of a POSITA to simply implement the additional aging process taught by Ishimori to Smith-Sharma-Edsall, and a POSITA would have been able to do so with a reasonable expectation of success. *Id.*

4. Smith, Sharma, Ishimori, and Zelig

110. A POSITA would have been motivated to combine Smith-Sharma-Ishimori with Zelig. The Smith-Sharma-Ishimori combination would have been a virtual network device with several virtual network device sub-unit operating as a single logical network device with MAC bridges. Ex. 1004 ¶ 54; Ex. 1026 3:42-50;

Ex. 1005 ¶ 9; Ex. 1007 at Abstract. It would have been obvious to a POSITA to implement the MAC bridges in Smith-Ishimori to maintain a separate MAC table for each VPN MAC bridge as taught in Zelig.

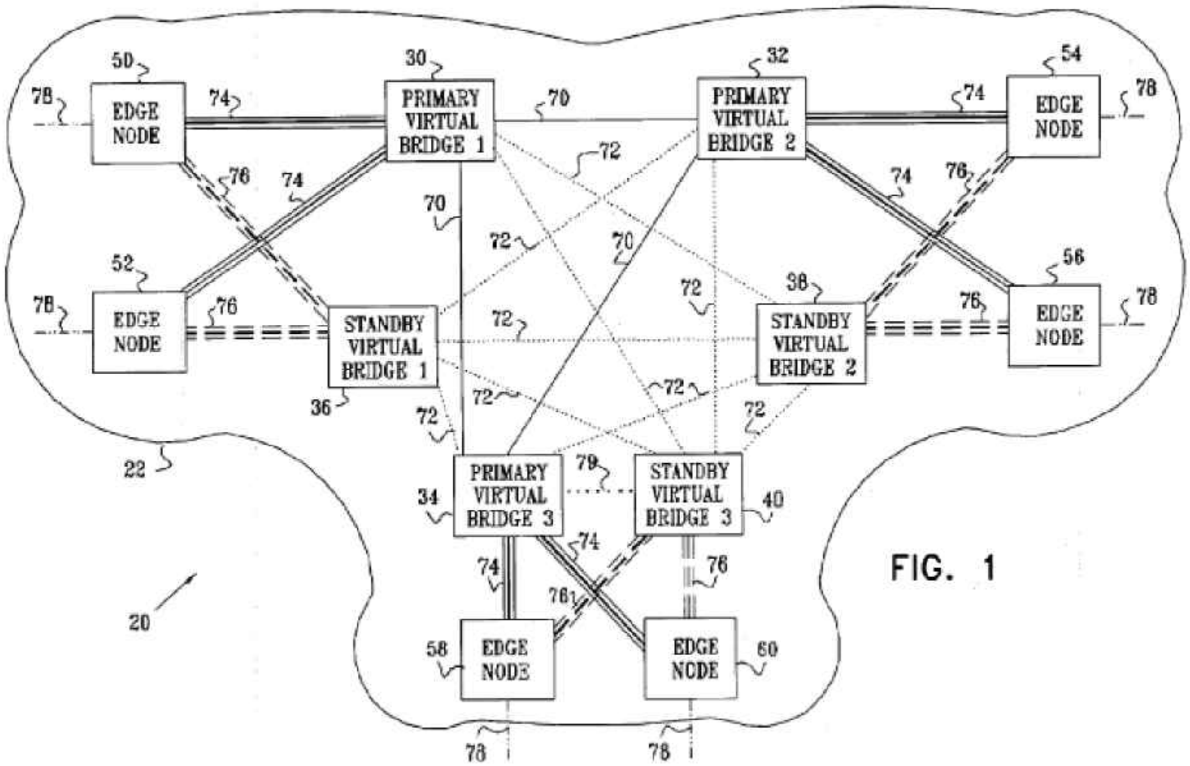


FIG. 1

111. It would have also been obvious to a POSITA to configure each virtual MAC bridge to serve a respective VPN, as shown above in Figure 1. Furthermore, implementing the MAC bridges as taught in Zelig to the Smith-Sharma-Ishimori data communication network would have been an implementation that a POSITA would have known how to make with a reasonable expectation of success because, for example, Zelig teaches how to apply the techniques to systems having similar structure as Smith-Sharma-Ishimori.

5. Smith, Sharma Ishimori, Zelig, and 802.1Q-1998

112. A POSITA would have been motivated to combine Smith-Sharma-Ishimori-Zelig with 802.1Q-1998. All references pertain to MAC bridges in a data communication network. Ex. 1004 ¶ 54; Ex. 1026 3:43-50; Ex. 1005 ¶ 9; Ex. 1007 at Abstract; Ex. 1008 at Abstract. Moreover, Zelig expressly cites to the 802.1Q standard. Ex. 1007 ¶ 12. A POSITA would have therefore been motivated to look to the teachings of 802.1Q-1998 in combination with Smith-Ishimori-Zelig to implement the messages from Smith-Sharma-Ishimori-Zelig with a VLAN identifier in order to identify the VPLS instance. Additionally, a POSITA would have been motivated to improve the commercial applicability of the combined system by modifying the Smith-Sharma-Ishimori-Zelig system to be compliant with the 802.1Q standard. Furthermore, modifying the Smith-Sharma-Ishimori-Zelig message to add the VLAN identifier would have been a modification that a POSITA would have known how to make with a reasonable expectation of success.

C. Specific Grounds of Invalidity

1. Ground 1: Claims 1, 3, 6, 11, 13, and 16 would have been obvious over the Smith-Sharma combination

113. I have analyzed claims 1, 3, 6, 11, 13, and 16 and in my opinion they would have been obvious over Smith in view of Sharma. I provide a detailed analysis of each claim limitation below.

a. Claim 1[pre]: A method for communication, comprising:

114. In my opinion, Smith discloses claim 1[pre]. Smith teaches a system that includes "several communication links." Ex. 1004 ¶ 9. As a stated in the background, computer communication networks are comprised of communication links (and nodes), and these communication links result in a method for communication. Indeed, Smith's disclosure "relates to networking." *Id.* ¶ 2.

b. Claim 1[a]: configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;

115. In my opinion, Smith discloses claim 1[a]. In figure 3 of Smith, there is a **virtual network device 202**, which is a network node. *Id.* ¶ 36. The virtual network device includes **several line cards**, such as 304(1) and 304(4), which may correspond to the first and second line cards. *Id.* ¶ 46. "In virtual network device sub-unit 122(1), line card 304(1) includes forwarding engine 314(1) and interfaces 320(5), 320(7), and 320(9)." *Id.* ¶ 47. "Line card 304(4) includes forwarding engine 314(4) and interfaces 320(12), 320(14), and 320(16)." *Id.* ¶ 48. The interfaces on line cards 304(1) and 304(4) are a plurality of ports. An "interface" and "port" are synonymous and Smith treats them interchangeably, and thus, discloses the first and second ports. *Id.* ¶ 63 (stating "port *or* uplink interface"). Therefore, it is my opinion that Smith's virtual network device, which is a network node, is configured to have a first and second line card with respective first and second ports.

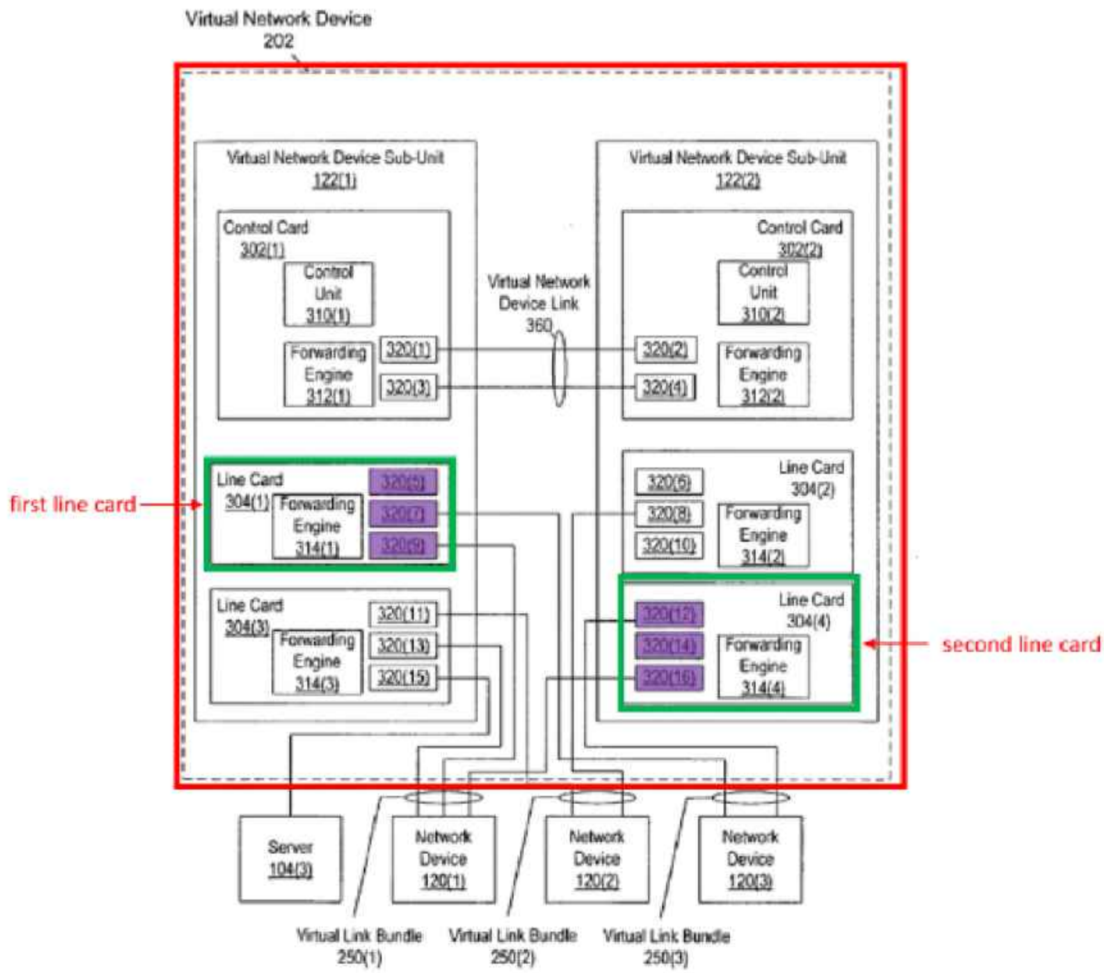


FIG. 3

116. Smith further teaches virtual link bundles, which may be link aggregation (LAG) groups and may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶ 6, 9, 30. Indeed, Smith discloses that:

a. "One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using a **link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**." *Id.* ¶ 6 (emphasis added).

b. "In some embodiments, virtual link bundles **250(1)** and **250(2)** are each operated as an EtherChannel (TM) or as an **aggregated link (as described in IEEE 802.3)**." *Id.* ¶ 35. (emphasis added).

c. "[...] the non-satellite network devices provide **L2 (Layer 2)** and L3 (Layer 3) **forwarding** and routing [...]" and "[...] the satellite network devices simply forward all packets to non-satellite network devices for **L2 forwarding** and L3 routing." *Id.* ¶ 30 (emphasis added).

d. "For a given virtual link bundle, that virtual link bundle can be managed (e.g., with respect to control protocols such as **L2 protocols**) in a central location." *Id.* ¶ 57 (emphasis added).

e. "In some embodiments, **MAC** notification frames are used to keep the content of the **L2 tables** in virtual network device sub-unit **122(1)** synchronized with the content of the **L2 tables** in virtual network device sub-unit **122(2)** and vice versa." *Id.* ¶ 62 (emphasis added).

f. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1)**) is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can learn that the sending device's **MAC (Media Access Control)** address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13)**. Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well as each

forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1)**) and **320(16)** (in virtual network device sub-unit **122(2)**) also have the same logical identifier as uplink interface **320(13)**, a **packet addressed to that MAC address can be forwarded** via any of uplink interfaces **320(9)**, **320(13)**, and **320(16)**." *Id.* ¶ 54 (emphasis added).

g. "The communication links are configured to be **managed as a single link.**" *Id.* ¶ 9 (emphasis added).

h. "Network devices **120(1)-120(3)** each operate their multiple uplinks to virtual network device **202** as a **single logical uplink**. Additionally, in some embodiments, each network device **120(1)-120(3)** can operate as if that network device is coupled to a **single distribution-layer device**, virtual network device **202**, instead of operating as if that network device were coupled to two independent distribution-layer network devices." *Id.* ¶ 44 (emphasis added).

117. The virtual network device 202 "route[s] and forward[s] packets to and from network devices 120(1)–120(3)" by associating the MAC address of a received data packet with the logical identifier of the uplink interface. *Id.* ¶ 54. Therefore, it is my opinion that Smith's virtual network device 202 operates as a distributed MAC bridge in Layer 2 of the data network.

118. In conclusion, it is my opinion that Smith teaches configuring a network node (the virtual network device) having at least a first and second line card (the line

cards 304(1) and 304(4)) with a plurality of ports (the interfaces on the line cards) that operates as a distributed MAC bridge in Layer 2 of the data network.

c. Claim 1[b]: configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined Member Line Cards;

119. In my opinion, Smith discloses claim 1[b]. Smith teaches virtual link bundles, which may be link aggregation (LAG) groups and may provide Layer 2 forwarding, and are "managed as a single link." *Id.* ¶¶ 6, 9, 30. Indeed, Smith discloses that:

a. "One way to avoid the complexity of having several independent redundant links is to operate those links as single logical transmission path, such as that provided using a **link bundling technique like** EtherChannel (TM) or **link aggregation (defined in IEEE 802.3)**." *Id.* ¶ 6 (emphasis added).

b. "In some embodiments, virtual link bundles **250(1)** and **250(2)** are each operated as an EtherChannel (TM) or as an **aggregated link (as described in IEEE 802.3)**." *Id.* ¶ 36 (emphasis added).

c. "[T]he non-satellite network devices provide **L2 (Layer 2)** and L3 (Layer 3) **forwarding** and routing" and "the satellite network devices simply forward all packets to non-satellite network devices for **L2 forwarding** and L3 routing." *Id.* ¶ 30 (emphasis added).

d. "For a given virtual link bundle, that virtual link bundle can be managed (e.g., with respect to control protocols such as **L2 protocols**) in a central location." *Id.* ¶ 57 (emphasis added).

e. "In some embodiments, **MAC** notification frames are used to keep the content of the **L2 tables** in virtual network device sub-unit **122(1)** synchronized with the content of the **L2 tables** in virtual network device sub-unit **122(2)** and vice versa." *Id.* ¶ 62 (emphasis added).

f. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1)**) is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can learn that the sending device's **MAC (Media Access Control)** address is "behind" uplink interface 320(13) by associating the MAC address with the logical identifier of uplink interface **320(13)**. Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device sub-unit **122(1)** as well as each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1)**) and **320(16)** (in virtual network device sub-unit **122(2)**) also have the same logical identifier as uplink interface **320(13)**, a **packet addressed to that MAC address can be forwarded** via any of uplink interfaces **320(9)**, **320(13),** and **320(16)**. *Id.* ¶ 54 (emphasis added).

g. "The communication links are configured to be **managed as a single link.**" *Id.* ¶ 9 (emphasis added).

h. "Network devices **120(1)-120(3)** each operate their multiple uplinks to virtual network device **202** as a **single logical uplink**. Additionally, in some embodiments, each network device **120(1)-120(3)** can operate as if that network device is coupled to a **single distribution-layer device**, virtual network device **202**, instead of operating as if that network device were coupled to two independent distribution-layer network devices." *Id.* ¶ 44 (emphasis added).

120. Smith discloses that **network device 120(2)** or one of the endpoints "is coupled to **virtual network device 202**" or a second endpoint "by **virtual link bundle 250(2)**" as shown in the annotated figure below. *Id.* ¶ 44. The virtual link bundle 250(2) consists of **two uplinks**, which is a plurality of LAG ports. *Id.* ¶ 51.

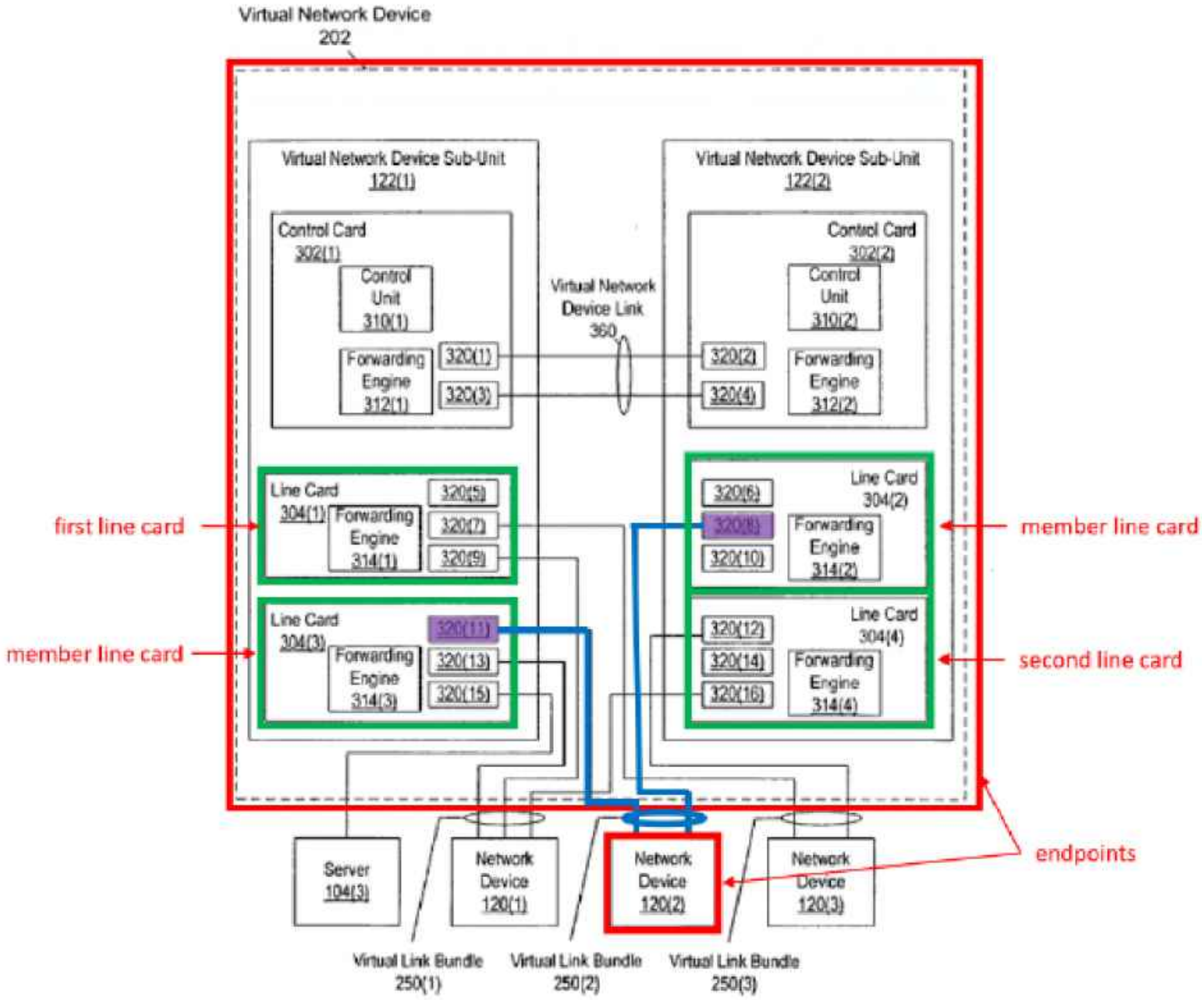


FIG. 3

121. Smith further specifies that the virtual network devices "provide L2 (Layer 2) . . . forwarding and routing." *Id.* ¶ 30. Continuing with my above example, line cards 304(1) and 304(4) may correspond to the first and second line card respectively. The other two line cards 304(2) and 304(3) may make up the conjoined member line cards, because all four line cards are part of the virtual network device sub-units 122(1) and 122(2) that "can coordinate their behavior such that they appear to be a single virtual device." *Id.* ¶ 50. In my opinion, Smith therefore discloses

configuring a link aggregation (LAG) group of parallel physical links (the virtual link bundle 250(2)) between two endpoints (virtual network device 202 and network device 120(2)) in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports (interfaces 320(8) and 320(11)) and a plurality of conjoined Member Line Cards (line cards 304(2) and 304(3)).

d. Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

122. In my opinion, Smith discloses claim 1[c]. The member line cards in Smith include **interfaces**, which are ports. Ex. 1004 ¶ 47. And the forwarding tables in Smith correspond to forwarding databases. Indeed, Smith discloses:

a. "Line card **304(3)** includes forwarding engine **314(3), interfaces 320(11)** and **320(13),** and **port 320(15)**. *Id.* ¶ 47 (emphasis added).
b. "identifier of the identified port or uplink interface . . . " *Id.* ¶ 65 (emphasis added).
c. "If the packet is received from a local uplink **interface or port** (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405**, the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can provide the destination address to a **forwarding table** in order to determine which **logical identifier, if any, is associated with that destination address**. If there is no hit in the **forwarding table**, as determined at **407**, the virtual network device sub-unit has

not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409**. For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

d. "If the packet's destination address hits in the **forwarding table**, as determined at **407**, the virtual network device sub-unit uses the logical identifier returned by the **forwarding table** to select the interface(s) to which the packet should be sent. If the **forwarding table** does not identify an interface bundle, as determined at **411**, the packet is sent via the identified port(s) and/or uplink interface(s), as indicated at **413**. If the **forwarding table** does identify an interface bundle, the virtual network device sub-unit sends the packet via one local interface included within the identified interface bundle, as shown at **415** (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

e. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the

packet's destination device. In this example, this is performed by providing the destination address to a **forwarding table**, as shown a **417**. If there is not a hit in the **forwarding table** (i.e., if no association has already been learned for the destination address), the virtual network device sub-unit floods the packet on the incoming VLAN." *Id.* ¶ 69 (emphasis added).

f. "If there is a hit in the forwarding table, and if the forwarding table does not identify an interface bundle at **421**, the packet is sent via the identified port and/or uplink interfaces, as indicated at **423**. If instead the forwarding table does identify an interface bundle, the packet is not sent via that interface bundle." *Id.* ¶ 70 (emphasis added).

g. "When virtual network device sub-unit **122(1)** looks up the destination address of the packet in a **lookup table**, the lookup table returns the logical identifier that identifies local uplink interfaces **320(9)** and **320(13)**. The packet is then forwarded to uplink interface 320(13) (e.g., selected based on load-sharing considerations)." *Id.* ¶ 61 (emphasis added).

h. "For example, control unit **310(2)** can use this information to set up or modify **lookup tables on line cards 304(2)** and/or **304(4)**." *Id.* ¶ 57 (emphasis added).

i. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can

learn that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13)**. Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13)**, and **320(16)**." *Id.* ¶ 54 (emphasis added).

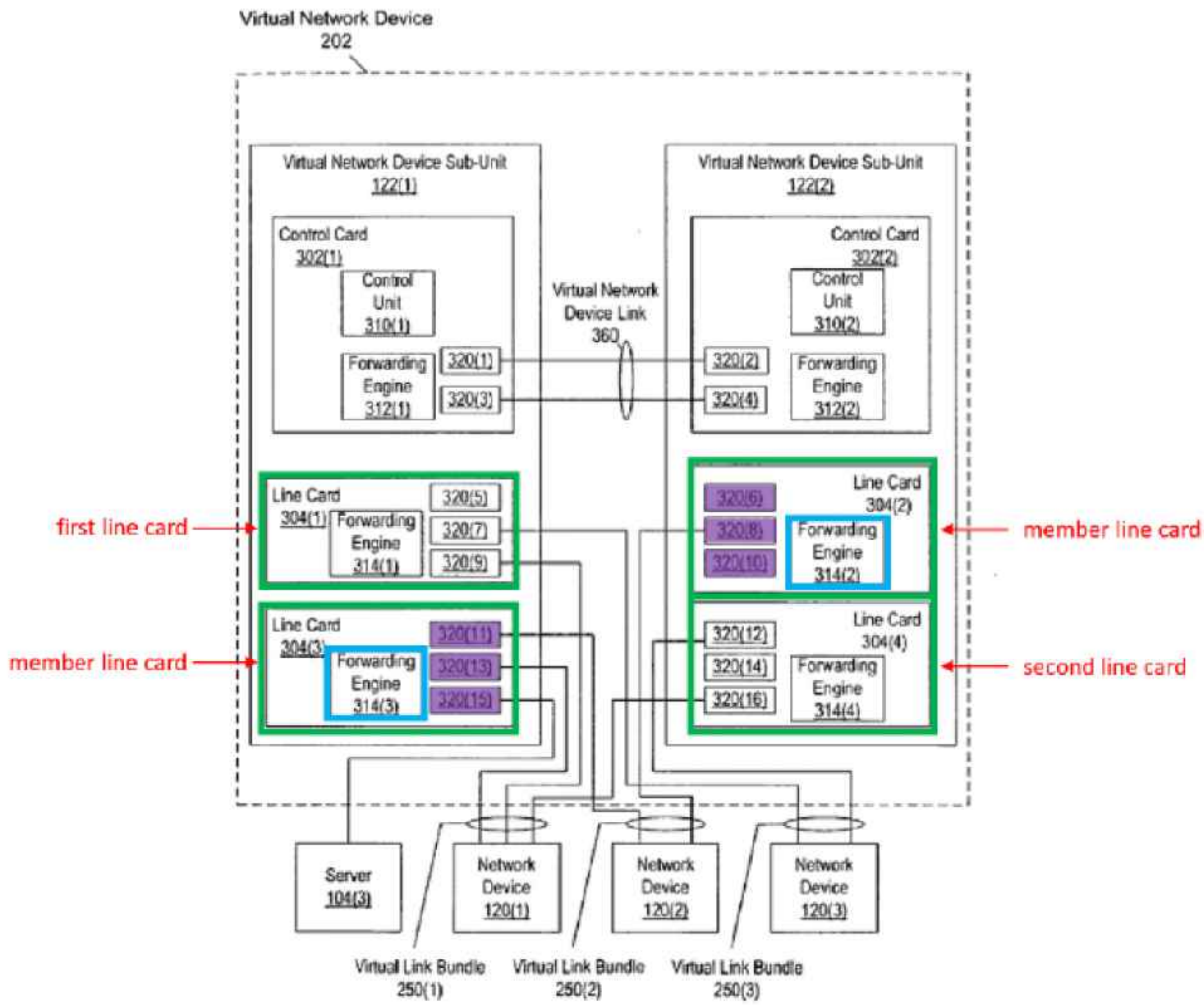


FIG. 3

123. When a packet is received on a particular uplink interface, the virtual network device learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] **uplink interface**" (e.g., associating MAC addresses with ports of said plurality of ports of said network node). *Id.* ¶54. For subsequent packets that are addressed to one of the learned MAC addresses, the "**forwarding engine**" uses the learned association to identify the appropriate logical identifier and routes/forwards the packet accordingly. *Id.* Smith teaches this

information may be used to "set up or modify lookup tables" (e.g., the forwarding database (FDB)). *Id.* ¶¶57,61. Moreover, FIG. 2 illustrates each of the **line cards** having a respective **forwarding engine**. Therefore, it is my opinion that Smith teaches that the lookup table or forwarding table or FDB may store records associating MAC addresses with the logical identifier of the uplink interface of said network node.

e. Claim 1[d]: receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;

124. In my opinion, Smith discloses claim 1[d]. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a **packet from a sending device (e.g., a client coupled to network device 120(1)) is received via uplink interface 320(13), virtual network device sub-unit 122(1) can learn that the sending device's MAC (Media Access Control) address** is "behind" uplink interface **320(13)** by associating the MAC address with the logical identifier of uplink interface **320(13)**. Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device subunit **122(1)** as well as each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, **packets addressed to that MAC address will be sent from an**

uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16)**." *Id.* ¶ 54 (emphasis added)

b. "If the packet is **received from a local uplink** interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405**, the virtual network device sub-unit attempts to **forward the packet to its destination address**. For example, the virtual network device sub-unit can provide the **destination address** to a forwarding table in order to determine which logical identifier, if any, is associated with that **destination address**. If there is no hit in the forwarding table, as determined at **407**, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409**. For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface

bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

125. Smith discloses that the uplink interface receives a data packet with the "sending device's MAC address." *Id.* ¶ 54. The uplink interface is an ingress port and the sending device's MAC address is the MAC source address. For example, if the network device 120(1) is the MAC source address, the ingress port would be interface 320(13) from a data packet received from network device 120(1).

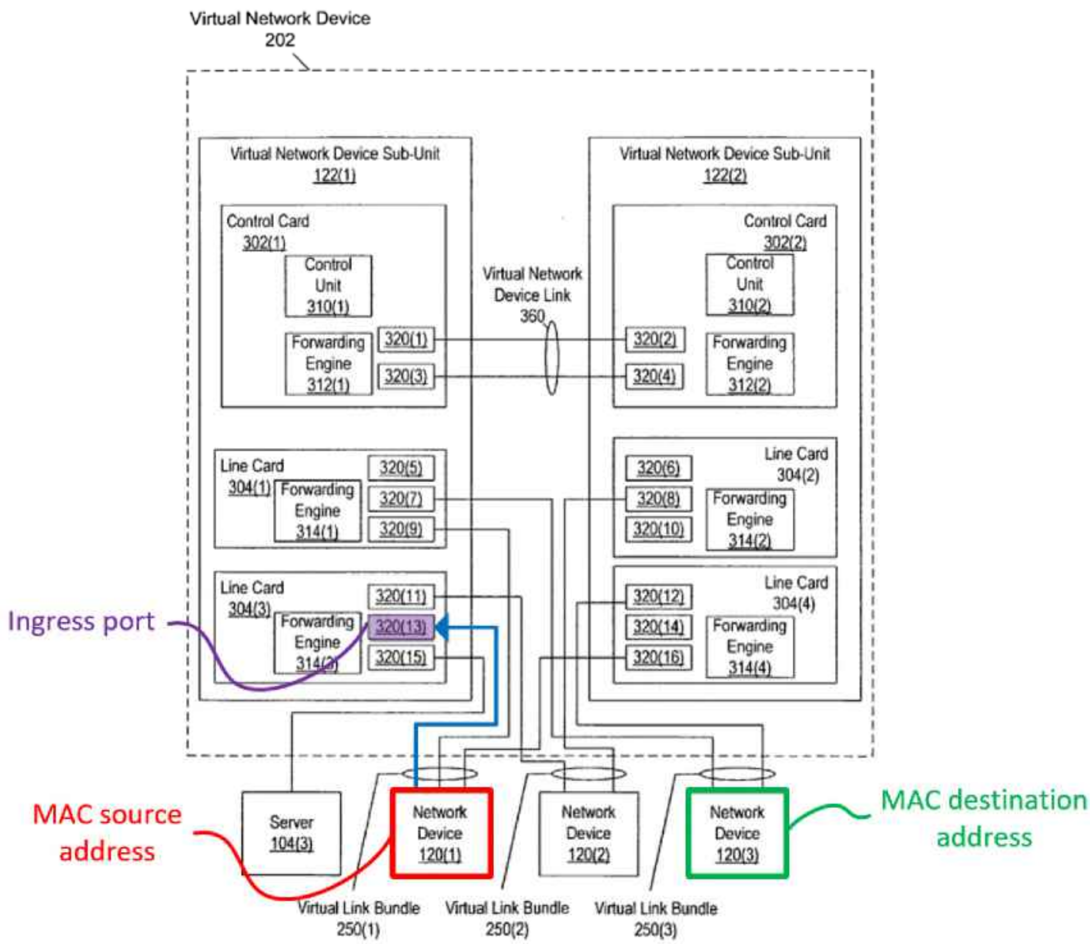


FIG. 3

126. It is therefore my opinion that Smith teaches receiving a data packet on an ingress port (interface 320(13)) of said network node (virtual network device 202) from a MAC source address (network device 120(1)).

127. Smith further discloses that this data packet has a "destination logical identifier." *Id.* ¶¶ 60, 62. The MAC destination address could be network device 120(3), for example. Smith specifies that these devices provide Layer 2 forwarding. *Id.* ¶ 30. The data packet's MAC destination address (network device 120(3)) is on the Layer 2 of the data network. In my opinion, Smith teaches receiving a packet on an ingress port (for example, interface 320(9)) of said network node from a MAC source address (for example, network device 120(1)), said data packet specifying a MAC destination address (for example, network device 120(3)).

f. Claim 1[e]: conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;

128. In my opinion, Smith discloses claim 1[e]. Smith discloses:

a. "The association between a packet and a particular logical identifier can be used by forwarding engines within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a **packet from a sending device (e.g., a client coupled to network device 120(1)) is received via uplink interface 320(13), virtual network device sub-unit 122(1) can learn that the sending device's MAC (Media Access Control) address** is "behind" uplink interface **320(13)** by

associating the MAC address with the logical identifier of uplink interface **320(13)**. Virtual network device sub-unit **122(1)** can inform each forwarding engine in virtual network device subunit **122(1)** as well **as** each forwarding engine in virtual network device sub-unit **122(2)** of this association. Based on the association, **packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier**. Since in this case, uplink interfaces 320(9) (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16)**." *Id.* ¶ 54 (emphasis added).

b. "If the packet is **received from a local uplink** interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405**, the virtual network device sub-unit attempts to **forward the packet to its destination address**. For example, the virtual network device sub-unit can provide the **destination address** to a forwarding table in order to determine which logical identifier, if any, is associated with that **destination address**. If there is no hit in the forwarding table, as determined at **407**, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the

packet's source address), excluding the interface that the packet arrived on, as shown at **409**. For **interfaces (e.g., ports or uplinks)** included in interface bundles, the virtual network device sub-unit selects one **egress interface** per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

129. Smith teaches conveying a packet to the appropriate uplink interface (port) based on the MAC destination address. EX1004 ¶54. Again, referring to Figure 3, in order to transmit the data packet to the MAC destination address (network device 120(3)), the data packet would be conveyed from interface 320(13) to interface 320(7) (said first port), because the system favors local interfaces. Interface 320(7), which is on line card 304(1) (said first line card), would then transmit the data packet to the MAC destination address (port of network device 120(3)).

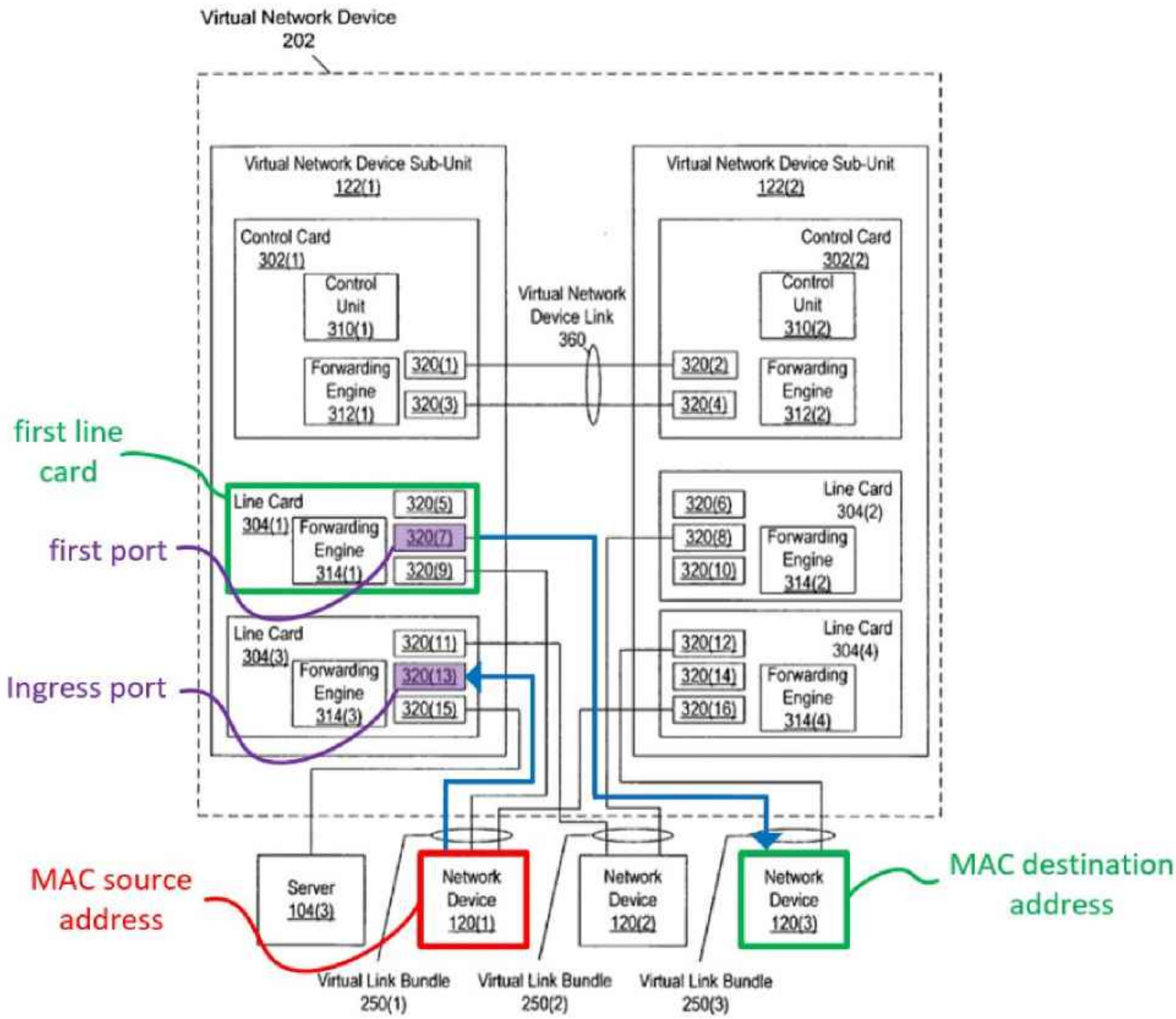


FIG. 3

130. Smith also teaches that particular interfaces may "act as ingress-only or egress-only." EX1004 ¶49. Accordingly, if a packet received on an interface designated as "ingress-only" (e.g., **ingress port, or interface 320(13)**), the packet would then have to be conveyed to another interface that could perform egress functions (e.g., **first port, or interface 320(7)**). Moreover, Smith teaches that its redundant architecture allows the network device 202 to continue operating even if

some of its interfaces were to fail. For example, if a packet is received on a particular interface that fails and is no longer able to output the packet, the packet would have to be conveyed to another interface that is able to output the packet. *Id.*

g. **Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;**

131. In my opinion, Smith discloses claim 1[f]. Smith teaches:

a. "If the packet is received from a local uplink interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at 405, the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can provide the destination address to a forwarding table in order to determine which logical identifier, if any, is associated with that destination address. **If there is no hit in the forwarding table**, as determined at 407, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit **floods the packet to all egress ports** and/or uplink interfaces in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at 409. For interfaces (e.g., ports or uplinks) **included in interface bundles**, the virtual network device sub-unit selects **one egress interface per interface bundle** via which to send the packet. If the packet was received by the virtual network device

sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

b. "If the packet's destination address hits in the forwarding table, as determined at 407, the virtual network device sub-unit uses the logical identifier returned by the forwarding table to select the interface(s) to which the packet should be sent. If the forwarding table does not identify an interface bundle, as determined at 411, the packet is sent via the identified port(s) and/or uplink interface(s), as indicated at 413. If the forwarding table does identify an interface bundle, the virtual network device sub-unit sends the packet via **one local interface** included within the identified interface bundle, as shown at 415 (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

c. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the packet's destination device. In this example, this is performed by providing the destination address to a forwarding table, as shown a **417. If there is not a hit in the forwarding table** (i.e., if no association has already been learned for the destination address), the virtual network device sub-unit **floods the packet on the** incoming VLAN." *Id.* ¶ 69 (emphasis added).

132. Smith teaches that the virtual network device sub-unit 122(1) looks up the destination address in a "lookup table" (FDB). *Id.* ¶ 61. Smith discloses sharing entries of the FDB with all the member line cards. *Id.* ¶ 63. "If a forwarding engine

within virtual network device sub-unit 122(2) already knows that the destination address is behind a particular uplink interface or port . . . , that forwarding engine generates a MAC notification identifying this association, which is distributed to any other forwarding engines within virtual network device sub-unit 122(2)." *Id.*

133. If there is no hit in the forwarding table, "the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces in the incoming VLAN." *Id.* ¶¶ 63, 66. Smith discloses that "flooding" is "via one and only one LAG port." In particular, Smith teaches that when the network device sends a packet "via the virtual link bundle," it "selects one of the communication links on which to send the packet." *Id.* ¶ 9. Sharma similarly discloses selecting one port within aggregated ports, through which frames may be transmitted. EX1026 1:57-59. Thus, it my opinion that Smith, alone or in combination with Sharma, discloses, or at least renders obvious, sending the data packet via one and only one LAG port.

h. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

134. In my opinion, Smith, alone or in combination with Sharma, discloses claim 1[g]. I provide my opinion for claim 1[g] with claim 1[h] below.

i. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.

135. In my opinion, Smith, alone or in combination with Sharma, discloses claims 1[h] and 1[g].

136. Consistent with the language of claim 1[e], the "first line card" recited in claims 1[g] and 1[h] is a line card that the data packet is conveyed to "for transmission to said MAC destination address," or otherwise referred herein as an "egress line card."

137. Sharma discloses a technique referred to as "egress learning," where the source address is learned on the line card at which the packet is to be transmitted out of the network device, rather than the line card at which the packet is first received by the network device. EX1026 4:6-23; Egress learning involves "search[ing] of the egress port's forwarding table" to check whether an entry corresponding to the source address of a frame (data packet) can be found. *Id.* If an entry exists, the entry is updated with "the identify of the ingress port from which the frame was received." *Id.* If an entry is not found, "a new entry is created and added to the table," which specifies "the SA and an identifier of the ingress port from which the egress port received the frame. *Id.* 4:24-32. Sharma further discloses that each of its ports have an associated forwarding table, and that each port can "reside on different line cards." *Id.* 3:45-50,5:11-14. Thus, Sharma discloses checking the MAC source address of the data packet against the FDB of the first line card (on the

egress port), and updating the FDB is the MAC source address and its association with the ingress port is not already in the FDB.

138. Sharma further discloses that entries in a forwarding table can be periodically synchronized with other forwarding tables. EX1026 5:11-21. Smith similarly teaches that when a particular forwarding engine learns of a new source address association with an ingress port, a MAC notification (e.g., a message) is sent to other forwarding engines to allow them to also learn the same. EX1004 ¶63. Thus, Smith and Sharma both disclose sending a message of the learned association to other line cards in the network device.

139. Smith teaches an alternative learning process where source address is learned on the line card at which the packet is first received by the network device:

a. "If the packet is received from a local uplink interface or port (i.e., if the packet is not received from another virtual network device sub-unit within the virtual network device), as determined at **405**, the virtual network device sub-unit attempts to forward the packet to its destination address. For example, the virtual network device sub-unit can **provide the destination address to a forwarding table** in order to determine which **logical identifier, if any, is associated with that destination address**. If there is no hit in the forwarding table, as determined at **407**, the virtual network device sub-unit has not yet learned the logical identifier of the interface(s) in front of the destination device(s). In this situation, the virtual network device sub-unit floods the packet to all egress ports and/or uplink interfaces

in the incoming VLAN (Virtual Local Area Network) (the incoming VLAN is the VLAN that includes the device identified by the packet's source address), excluding the interface that the packet arrived on, as shown at **409**. For interfaces (e.g., ports or uplinks) included in interface bundles, the virtual network device sub-unit selects one egress interface per interface bundle via which to send the packet. If the packet was received by the virtual network device sub-unit via an interface bundle, all interfaces in that interface bundle are excluded from sending the packet." *Id.* ¶ 66 (emphasis added).

b. If the packet's destination address **hits in the forwarding table**, as determined at **407**, the virtual network device sub-unit uses the **logical identifier returned by the forwarding table** to select the interface(s) to which the packet should be sent. If the forwarding table does not identify an interface bundle, as determined at **411**, the packet is sent via the identified port(s) and/or uplink interface(s), as indicated at **413**. If the forwarding table does identify an interface bundle, the virtual network device sub-unit sends the packet via one local interface included within the identified interface bundle, as shown at **415** (if the forwarding table identifies other non-interface-bundle interfaces, the packet is sent via those interfaces as well)." *Id.* ¶ 67 (emphasis added).

c. "The virtual network device sub-unit determines whether that sub-unit has already learned the logical identifier associated with the packet's destination device. In this example, this is performed by **providing the destination address to a forwarding table**, as shown a **417**. If there is not a hit in the forwarding table (i.e., if no

association has already been learned for the destination address), the virtual network device sub-unit floods the packet on the incoming VLAN." *Id.* ¶ 69 (emphasis added).

d. "If there is a **hit in the forwarding table**, and if the forwarding table does not identify an interface bundle at **421**, the packet is sent via the identified port and/or uplink interfaces, as indicated at **423**. If instead the forwarding table does identify an interface bundle, the packet is not sent via that interface bundle." *Id.* ¶ 70 (emphasis added).

e. "When virtual network device sub-unit **122(1) looks up the destination address of the packet in a lookup table**, the lookup table returns the logical identifier that identifies local uplink interfaces **320(9)** and **320(13)**. The packet is then forwarded to uplink interface **320(13)** (e.g., selected based on load-sharing considerations)." *Id.* ¶ 61 (emphasis added).

f. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13)**. Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit

122(2) of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16)**." *Id.* ¶ 54 (emphasis added).

g. "In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device subunit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification** can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via** to virtual network device link **360**. Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will send a copy of that packet via virtual network device link **360**, ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

h. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or

uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device subunit **122(1)** sends a copy of the packet to virtual network device sub-unit **122(2)** via virtual switch link **360**. If a forwarding engine within virtual network device sub-unit **122(2)** already knows that the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120**), that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device subunit **122(2).** Since the packet was originally received via virtual network device link **360**, virtual network device sub-unit **122(2)** also sends a copy of the MAC notification back via virtual network device link **360**. This MAC notification can then be distributed among the forwarding engines included in virtual network device sub- unit **122(1)**. After being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently- received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

140. Smith teaches that "[biased on the source address of the packet and which port or uplink interface received the packet, the virtual network device subunit learns the source identifier of the sending device, as indicated at 403." *Id.* ¶ 65. These identifiers are stored in a lookup table on a virtual network device. *Id.* ¶ 61.

It would have therefore been obvious to check the MAC source address in the records of the FDB of said first line card (line card 304(1)) to determine if it exists in the FDB.

141. A POSITA would have understood that if the record of an association between the MAC source address with said ingress port did not exist in the lookup table of, for example, line card 304(1) (said first line card), it would create a new entry and add it to the lookup table of line card 304(1). Smith teaches that the network device sub-unit 122(2) sends a MAC notification (a message) to update the forwarding engines when it learns of a new association. *Id.* ¶ 63. "After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand this to mean that the MAC notification, which is a message, is therefore sent to each member line card (304(2) through 304(4)).

j. Claim 3: The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

142. In my opinion, Smith discloses claim 3. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device

202 to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13)**. Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2))** also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16)**." *Id.* ¶ 54 (emphasis added)

b. "In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device subunit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification** can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via**

to virtual network device link **360**. Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will send a copy of that packet via virtual network device link **360**, ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

c. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device subunit **122(1)** sends a copy of the packet to virtual network device sub-unit **122(2)** via virtual switch link **360**. If a forwarding engine within virtual network device sub-unit **122(2)** already knows that the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120**), that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device subunit **122(2)**. Since the packet was originally received via virtual network device link **360**, virtual network device sub-unit **122(2)** also sends a copy of the MAC notification back via virtual network device link **360**. This MAC notification can then be distributed among the forwarding engines included in virtual network device sub- unit **122(1)**. After being updated based on the MAC notification, the forwarding

> engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently- received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

143. Smith discloses that the network device sub-unit 122(2) sends a MAC notification (the message) to update the forwarding engines. *Id.* ¶ 63. Smith specifies that "[a]fter being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand that this includes the other line cards in virtual network device 202, such as line card 304(4). For example, line card 304(4) (*e.g.*, the second line card) would receive the MAC notification (the message) and update its lookup tables. *Id.* ¶ 57. Thus, Smith discloses that in response to the message, the record of association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

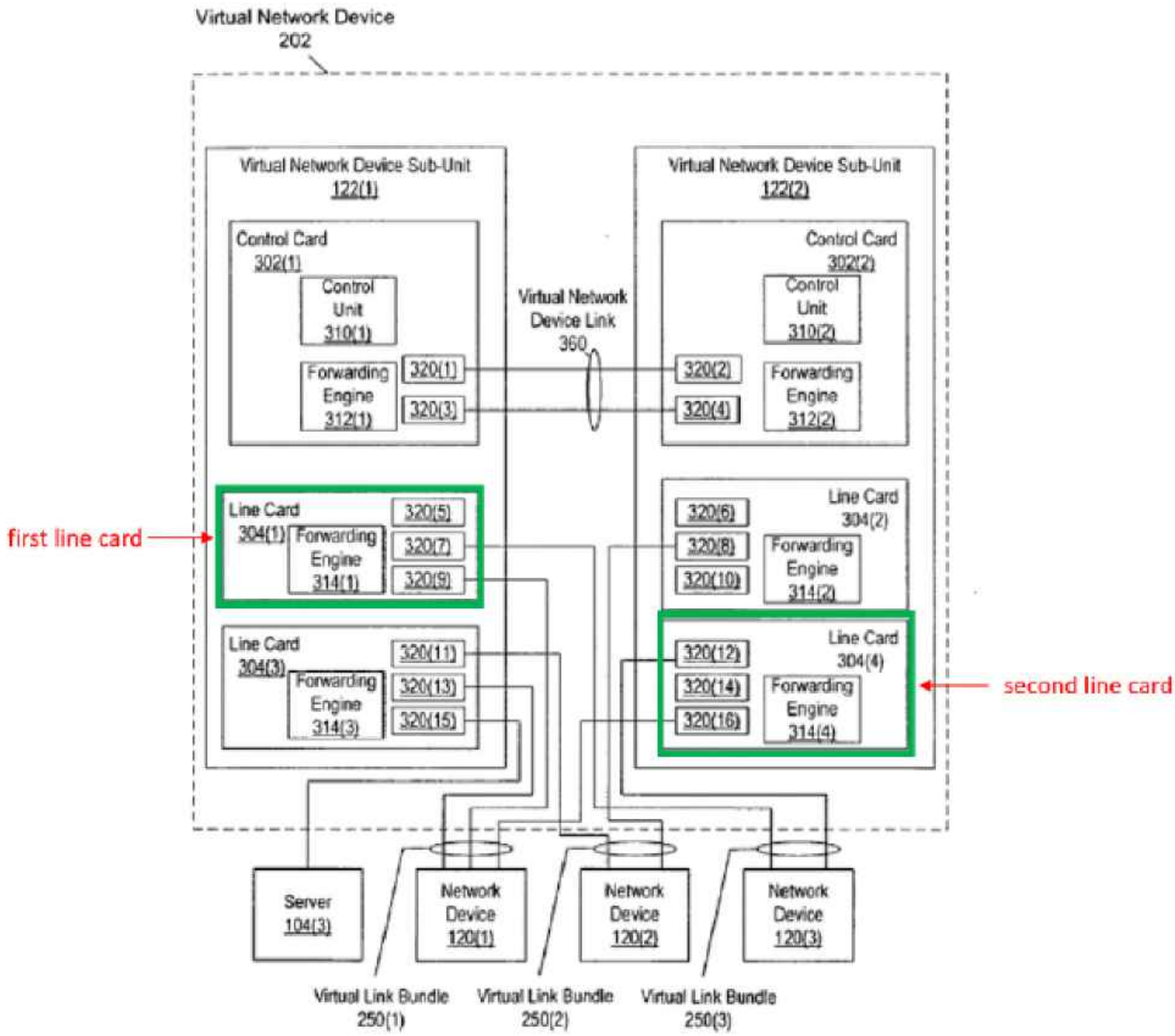


FIG. 3

144. Sharma discloses that entries in a forwarding table can be periodically synchronized with other forwarding tables. Ex. 1026 5:11-21.

k. Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via switching core of the network node to at least the second line card.

145. In my opinion, Smith discloses claim 6. Smith teaches:

a. "The association between a packet and a particular logical identifier can be used by **forwarding engines** within virtual network device **202** to route and forward packets to and from network devices **120(1)-120(3)**. For example, when a packet from a sending device (e.g., a client coupled to network device **120(1))** is received via uplink interface **320(13)**, virtual network device sub-unit **122(1)** can **learn** that the sending device's MAC (Media Access Control) address is "behind" uplink interface **320(13)** by **associating the MAC address with the logical identifier** of uplink interface **320(13)**. Virtual network device sub-unit **122(1) can inform each forwarding engine** in virtual network device sub-unit **122(1)** as well **as each forwarding engine** in virtual network device sub-unit **122(2)** of this association. Based on the association, packets addressed to that MAC address will be sent from an uplink interface having the associated logical identifier. Since in this case, uplink interfaces **320(9)** (in virtual network device sub-unit **122(1))** and **320(16)** (in virtual network device sub-unit **122(2)**) also have the same logical identifier as uplink interface **320(13)**, a packet addressed to that MAC address can be forwarded via any of uplink interfaces **320(9), 320(13),** and **320(16)**." *Id.* ¶ 54 (emphasis added).

b. "In some embodiments, **MAC notification frames** are used to keep the **content of the L2 tables** in virtual network device subunit **122(1) synchronized** with the content of the L2 tables in virtual network device sub-unit **122(2)** and vice versa. Whenever a MAC notification that involves a port behind a virtual link bundle or an uplink interface included in an uplink interface bundle is generated **within a virtual network device sub-unit** (e.g., **such a notification**

can be generated **by one line card** in order to **update an L2 table on another line card**), a **copy of the MAC notification is sent via** to virtual network device link **360**. Similarly, if a virtual network device sub-unit determines that a packet should be flooded, the virtual network device sub-unit will send a copy of that packet via virtual network device link **360**, ensuring that the virtual network device sub-unit will receive a copy of any MAC notification response generated by a forwarding engine in the peer virtual network device sub-unit." *Id.* ¶ 62 (emphasis added).

c. "By way of example, assume that virtual network device sub-unit **122(1)** floods a packet because the forwarding engine(s) included in virtual network device sub-unit **122(1)** do not know which port or uplink interface is associated with the packet's destination address. As part of flooding the packet, virtual network device subunit **122(1)** sends a copy of the packet to virtual network device sub-unit 122(2) via virtual switch link **360.** If a forwarding engine within virtual network device sub-unit **122(2)** already knows that the destination address is behind a particular uplink interface or port (e.g., if a **forwarding table already includes an entry associating the destination address with a port** of one of network devices **120**), that forwarding engine **generates a MAC notification** identifying this association, which is **distributed to any other forwarding engines** within virtual network device sub-unit **122(2)**. Since the packet was originally received via virtual network device link **360**, virtual network device sub-unit **122(2)** also sends a copy of the MAC notification back via virtual network device link **360**. This MAC notification can then be distributed among the forwarding

engines included in virtual network device sub- unit **122(1)**. After being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit **122(1)** now know the location of the device identified by the destination address. Accordingly, subsequently-received packets addressed to that device will not be flooded." *Id.* ¶ 63 (emphasis added).

146. The line cards communicate with each other in order to transfer various messages between them. For example, note that "such a notification can be generated by one line card in order to update an L2 table on another line card." *Id.* ¶ 62. Therefore, the line cards must communicate, allowing for sending messages from one line card to another. This, internal to device 202, switching mechanism allowing the line cards to communicate with each other could be the switching core.

147. Smith teaches that the "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* Thus, it would have been obvious to a POSITA to implement the MAC notification frame as a synchronization packet and transmit it to at least the second line card via the switching core.

l. Claim 11[pre] 11: A node for network communication, comprising:

148. In my opinion, Smith discloses claim 11[pre]. Smith teaches that its system contains communications links that are coupled to network devices. *Id.* ¶ 9. A POSITA would understand that a network device is a node.

m. Claim 11[a]: a switching core;

149. In my opinion, Smith discloses claim 11[a]. Smith discloses a switching mechanism that allows line cards to communicate with each other. EX1004 ¶62. A switching core was a well-known technique for line cards to communicate with one another in a switching device. It would be within the knowledge of a POSITA that the line cards in Smith would be communicating with each other via a switching core, as discussed before. *See supra* ¶ 146.

n. Claim 11[b]: a plurality of ports;

150. In my opinion, Smith discloses claim 11[b]. The network devices in Smith contain several ports (a plurality of ports). Ex. 1004 ¶ 6. "For example, an EtherChannel (TM) port bundle can be formed ***from several ports on a switch***, each of which is coupled to a respective link in a group of links coupling that switch to another switch." *Id.*

o. Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of

member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

151. In my opinion, Smith, either alone or in combination with Sharma, discloses claim 11[c]. Claim 11[c] combines the elements of claims 1[a], 1[b], 1[c], and 11[a]. Therefore, for the same reasons that the Smith-Sharma combination discloses claims 1[a], 1[b], 1[c], and 11[a], the Smith-Sharma combination also discloses claim 11[c]. *See supra* ¶¶ 112-120, 146.

p. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

152. In my opinion, Smith, either alone or in combination with Sharma, discloses claim 11[d]. This limitation is a combination of claims 1[d], 1[e], 1[f], 1[g], and 1[h]. Therefore, for the same reasons that the Smith-Sharma combination

discloses claims 1[d], 1[e], 1[f], 1[g], and 1[h], the Smith-Sharma combination also discloses claim 11[d]. *See supra* ¶¶ 121–135.

q. Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.

153. In my opinion, Smith, either alone or in combination with Sharma, discloses claim 13. *See supra* ¶¶ 139–140.

r. Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.

154. In my opinion, Smith, either alone or in combination with Sharma, discloses claim 16. *See supra* ¶¶ 141–143.

2. Ground 2: Claims 1–3 and 11–13 would have been obvious over the Smith-Sharma-Ishimori combination

155. I have analyzed claims 1–3 and 11–13 and conclude that they would have been obvious over Smith in combination with Sharma and Ishimori. I provide a detailed analysis of each claim limitation below.

a. Claim 1[c]: providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;

156. In my opinion, the combination of Smith, Sharma, and Ishimori render obvious claim 1[c]. As explained above, when a packet is received on a particular

uplink interface, the virtual network device learns the sending device's MAC address by "associating the MAC address with the logical identifier of [the] uplink interface." Ex. 1004 ¶ 54. Smith teaches that the information learned about the association between a packet and a particular logical identifier may be used to "set up or modify lookup tables" (*e.g.*, the forwarding database (FDB)). *Id.* ¶¶ 54, 57, 61, 71.

157. To the extent Smith or Sharma does not render obvious claim 1[c], Smith in combination with Sharma and Ishimori does. Indeed, Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of learning a source MAC address of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets, an IP packet having an unlearned MAC address as a destination address, a transmission path thereof cannot be specified. As such, in this situation, a broadcast transmission to all nodes (that is, "flooding") is performed. Then, at this time, path **learning** is carried out **by storing** the source MAC address (SA) had by this IP packet and a port that received such in a buffer had by each node." Ex. 1005 ¶ 2 (emphasis added).

b. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is

supposed wherein each node—50-0, 50-1, 50-2— has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A", which is a source MAC address (SA) thereof, is learned by being **stored** in a predetermined buffer in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the **MAC table**, this information is overwritten." *Id.* ¶ 3 (emphasis added).

c. "Next, for a destination address (DA; here, MAC address "B") had by this receive IP packet, it is searched whether there is a learning result in the MAC table of the local device (step S3). Then, when the search result is a mishit ("No"), this IP packet is broadcast (flooded) to all nodes (in this example, node 50-1 and node 50-2) (steps S5, P2). Meanwhile, when, as a result of the search at step S3, a learning result relating to the corresponding destination MAC address exists in the buffer ("Yes"), this IP packet is transmitted according to a forwarding path—that is, a card number and port number—included in this learning result (step S4). In this situation, there is no need for flooding." *Id.* ¶ 4 (emphasis added).

d. "Note that the same learning operations are also performed when this IP packet sent from node 50-0 reaches card #2 of node 50-2, which the IP packet passes through on its way to a corresponding destination terminal B 20. As a result, as illustrated in FIG. 1, source path information at this time—that is, the information on card #0, port #0 of node 50-0—is stored as a learning result in association

with the MAC address "A" of the source terminal 10 **in the buffer** (**MAC table**) of this card #2." *Id.* ¶ 5 (emphasis added).

158. Ishimori teaches that each of its communication cards has a "MAC table." *Id.* ¶ 2. Referring to Figure 1, Ishimori explains that when a packet from terminal A is received at port #0 of card #0, the path information for address "A" is learned by storing the information in the MAC table "had by this card" (*e.g.*, card #0). *Id.* ¶ 3. Ishimori therefore teaches that its line cards maintain a MAC table to store path information of the MAC address and the receiving card and port. *Id.* In other words, the MAC table holds records associating MAC addresses with ports of said plurality of ports of said network node. It would have been obvious to substitute the line cards from Ishimori into Smith and/or Sharma and a POSITA would have had a reasonable expectation of success in making this simple substitution.

b. Claim 1[f]: if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;

159. In my opinion, to the extent Smith or Sharma does not explicitly teach the limitation of sending the packet to one and only one LAG port, Ishimori expressly discloses flooding via "one representative port" (*e.g.*, via one and only one LAG port).

a. "Next, a **link aggregation** function is described. This link **aggregation function is a function of using a plurality of ports by deeming such to be one virtual high-speed port**. This function enables a band improvement effect to be substantially obtained. The

virtual port in this situation is referred to as a trunk. In using this trunk (referred to as "trunking"), **when, as above, flooding is to be performed when an address is not learned, one representative port is selected** from among the large number of ports had by this trunk according to a predetermined computation, and **actual packet transmission is performed using this representative port**." *Id.* ¶ 13 (emphasis added).

b. "Then, at a node that receives an IP packet via the trunk, when carrying out path learning of a source address of this packet, instead of learning the port number at the time of receipt, information on the trunk including such is stored as the learning result. Afterward, in receiving an IP packet having a destination that passes through this trunk, one path is decided from among the paths constituting the trunk according to predetermined computation methods, independent of each other according to hardware, of these communication cards, and this decided path is applied to the corresponding packet. This configuration **prevents the application of a different path in the same trunk to the same packet**." *Id.* ¶ 14 (emphasis added).

160. Ishimori teaches that when "flooding is to be performed when an address is not learned, one representative port is selected from among the large number of ports had by this trunk according to a predetermined computation, and actual packet transmission is performed using this representative port." *Id.* ¶ 13. One of ordinary skill in the art would have been motivated to combine the teachings of Smith with Ishimori because Ishimori explains that using "one representative port .

. . prevents the application of a different path in the same trunk to the same packet." *Id.* ¶¶ 13, 14. Thus, a POSITA would have looked to Smith, Sharma, and Ishimori, which explain why it would be beneficial to use only one representative port, and combine the teachings of the two references as they are directed to similar methods. Furthermore, a POSITA would have had a reasonable success in combining Smith and Sharma with Ishimori because it would have been a simple application of Ishimori's methods on Smith's, or Sharm's, virtual network device. Thus, the combination of Smith, Sharma, and Ishimori renders obvious claim 1[f].

c. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

161. In my opinion, to the extent this is not disclosed by Smith or Sharma, Ishimori teaches that the source MAC address is stored with the reception path information including the card number and port number. *Id.* ¶ 3. Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of **learning a source MAC address** of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets, an IP packet having an unlearned MAC address as a destination address, a transmission path thereof cannot be specified. As such,

in this situation, a broadcast transmission to all nodes (that is, "flooding") is performed. Then, at this time, path **learning** is carried out **by storing** the source MAC address (SA) had by this IP packet and a port that received such in a buffer had by each node." *Id.* ¶ 2 (emphasis added).

b. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is supposed wherein each node—50-0, 50-1, 50-2— has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A", which is a source MAC address (SA) thereof, **is learned** by being **stored** in a predetermined buffer in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the **MAC table**, this information is overwritten." *Id.* ¶ 3 (emphasis added).

162. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.* Thus, the combination of Smith, Sharma, and Ishimori renders obvious this limitation.

d. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB

> **of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.**

163. In my opinion, to the extent Smith or Sharma does not render obvious this limitation, in my opinion, it would have been obvious to combine Smith and Sharma with Ishimori. Ishimori discloses that the source MAC address is stored with the reception path information including the card number and port number. *Id.* ¶ 3. Indeed, Ishimori's disclosures include:

a. "FIG. 1 and FIG. 2 illustrate this. Note that in the example described below that is illustrated in FIG. 1 and the like, a configuration is supposed wherein each node—50-0, 50-1, 50-2— has one communication card—#0, #1, #2—respectively. When node 50-0 receives an IP packet from terminal A 10 at port #0 of card #0 (steps S1, P1), in a **MAC table** had **by this card**, "A", which is a source MAC address (SA) thereof, is learned by being **stored in a predetermined buffer** in association with information on the receiving card #0 and port #0 at this time (path information) (step S2). Note that in this situation, when a learning result for the same MAC address already exists in the MAC table, this information is overwritten." *Id.* ¶ 3 (emphasis added).

b. "The path having this representative port leads to card #3 of the nodes 70. At this card #3 as well, no learning result relating to the terminal 20 that is the destination is had. As such, at card #3 as well, this packet is flooded (step P13). In this manner, this packet reaches the terminal 20 that is the destination through card #5. Note that during the packet forwarding from the terminal 10 to the terminal

20 illustrated in FIG. 9 and FIG. 10, at each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16.

c. "According to the present invention, each packet forwarding device is configured to **generate a learn packet at a predetermined timing under predetermined conditions**. That is, even in a situation wherein different paths are selected for a coming direction and a going direction due to trunking or the like and thus only forwarding in one of these directions is performed in a certain device, by generating a learn packet at a predetermined timing, this learn packet can force, for example, the device performing forwarding in only one direction in this manner to also perform packet reception in the other direction. This causes each path to perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably. As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." *Id.* ¶ 25 (emphasis added).

d. "However, because an object of this learn packet is simply for effective path learning to be performed, generating this learn packet with needless frequency actually invites increased line traffic. To prevent such adverse effects, learn-packet generation conditions—that is, a generation frequency, a generation opportunity, and the like—must be appropriately determined." *Id.* ¶ 26.

e. "That is, it is desirable for the learn-packet generation conditions to be a situation wherein the path leading to the destination of the

receive packet includes a trunk, the learn packet being generated for the first time not at the time of the first learning but when the learning result 30 becomes a hit in a destination search. That is, a configuration is desirable that generates the learn packet when a packet is received whose destination is the learned source. Note that at this time, it is desirable to transmit the learn packet to all forwarding devices constituting this trunk. Moreover, it is desirable to also generate the learn packet each subsequent time when the learning result again becomes a hit in the destination search." *Id.* ¶ 28.

f. "To facilitate description, a situation is then supposed wherein the paths of cards #1, #3 are selected as the representative ports of the trunk in the direction wherein the terminal 20 receives 20 and, in contrast, the paths of cards #4, #2 are selected as the representative ports of the same trunk in the direction of transmitting to the terminal 10. In this situation, when no learn packet is generated, as described in conjunction with FIG. 12 to FIG. 14, flooding occurs repeatedly. That is, when a packet of a transmission direction toward the terminal 20 again reaches card #3, because card #3 has not received a packet of a reception direction from the terminal 20, no corresponding learning result is had, and flooding is again performed. However, according to this embodiment of the present invention, as above, the learn packet is sent from card #5 to cards #3, #4. As such, at both card #3 and card #4, the MAC address "B" of the terminal 20 is already learned. As such, even when a packet of a transmission direction toward the terminal 20 again reaches card

> #3, card #3 can specify the path to the terminal 20 according to the learning result provided by the learn packet." *Id.* ¶ 34.

164. It would have been obvious to a POSITA to check the MAC source address against the records of the MAC table to determine whether to add a new record, or as Ishimori discloses, to overwrite the record if the MAC address already exists in the MAC table. *Id.* Ishimori teaches a learning process where "each packet forwarding device is configured to generate a learn packet at a predetermined timing under predetermined conditions." *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33. Ishimori therefore discloses sending a message of the association to each member line card for said plurality of member line cards. It would have been obvious to implement the MAC notification in Smith to perform the method taught in Ishimori to first check whether the MAC address is found in the MAC table, and if not, create a new record of the association. Both Smith and Ishimori teach that a message of this association is sent to the plurality of member line cards. Ex. 1004 ¶ 63; Ex. 1005 ¶¶ 25, 33. Sharma also discloses this. EX1026 5:11-21. Thus, the combination of Smith, Sharma, and Ishimori renders obvious this limitation.

e. Claim 2: The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC address and the respective ports.

165. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 2. Smith teaches that the network device sub-unit 122(2) sends a MAC notification (a message) to update the forwarding engines, but it does not disclose doing so periodically at predefined times. Ex. 1004 ¶ 63. However, Ishimori discloses this. Indeed, Ishimori's disclosures include:

a. "According to the present invention, each packet forwarding device is configured to generate a **learn packet** at a **predetermined timing** under predetermined conditions. That is, even in a situation wherein different paths are selected for a coming direction and a going direction due to trunking or the like and thus only forwarding in one of these directions is performed in a certain device, by generating a learn packet at a **predetermined timing**, this learn packet can force, for example, the device performing forwarding in only one direction in this manner to also perform packet reception in the other direction. This causes each path to perform packet reception in both directions as appropriate, and path learning in both directions is performed reliably. As a result, performing flooding repeatedly and thus inviting increased line traffic can be prevented." Ex. 1005 ¶ 25 (emphasis added).

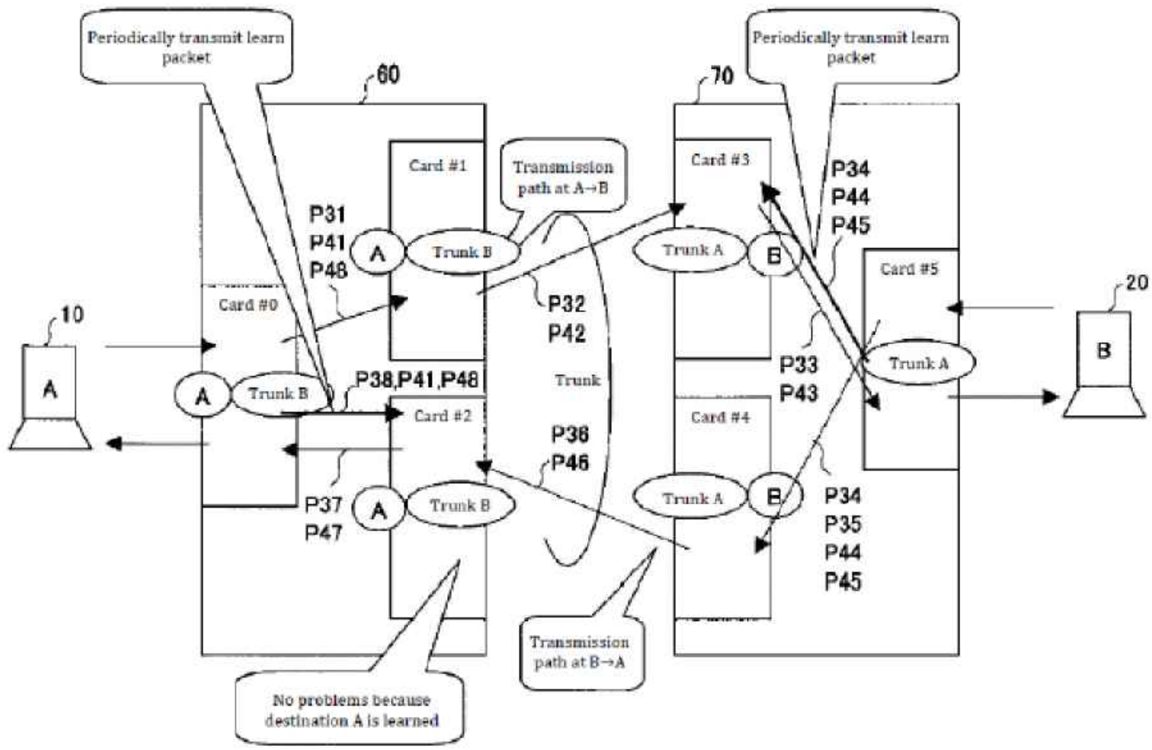
b. "In Figure 15, the path to the destination of the receive packet includes a trunk. Moreover, when, at these nodes, a packet from the same address as the corresponding destination address (DA) is already received and the source-address learning result from this time is had—that is, when there is a hit in the destination search (step S3 in FIG. 2)—a learn packet is transmitted **to all nodes having the**

corresponding trunk. That is, in FIG. 15, a packet is first forwarded from the terminal to the terminal 20. When, at each node that this forwarding operation passes through—for example, card #5—a reply packet from the terminal 20 to the terminal 10 is afterward received, at card #5, the learn packet is transmitted to all nodes having the corresponding trunk—that is, cards #3, #4. This causes the path for the corresponding MAC address to be learned at all of these nodes. That is, in this situation, at cards #3, #4, the path having card #5 is learned for the MAC address "B" of the terminal 20." *Id.* ¶ 33 (emphasis added).

c. "[Problem] An object is to provide, as a packet forwarding system that uses a path learning function applying a link aggregation function, a system that can prevent repeated flooding even when paths differ between a time of transmission and a time of reception. [Resolution Means] A configuration is adopted of periodically sending, according to predetermined conditions, a learn packet to all nodes having a trunk." *Id.*, Abstract.

d. Figure 15 shows "Periodically transmit learn packet":

[FIG. 15]
Block diagram for describing operations of packet forwarding system of embodiment of present invention



Id., Fig. 15.

166. Ishimori teaches that a "learn packet" (message) is generated at a "predetermined timing" (periodically at predefined times). *Id.* ¶ 25. Ishimori further teaches that the "learn packet" that informs the plurality of line cards regarding new associations "is transmitted to all nodes having the corresponding trunk." *Id.* ¶ 33. Therefore, it would have been obvious that this must include at least the second line card. *Id.* Thus, the combination of Smith, Sharma, and Ishimori discloses this limitation.

f. Claim 3: The method according to claim 1, and comprising receiving the message at the second line

card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

167. In my opinion, the combination of Smith and Ishimori discloses claim 3. Smith discloses that the network device sub-unit 122(2) sends a MAC notification (*e.g.*, the message) to update the forwarding engines. Ex. 1004 ¶ 63. Smith specifies that "[a]fter being updated based on the MAC notification, the forwarding engines in virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* I understand that this would include the other line cards in virtual network device 202, such as line card 304(4). For example, line card 304(4) (the second line card) would receive the MAC notification (the message) and update its lookup tables. *Id.* ¶ 57. Thus, Smith discloses that in response to the message, the record of association is added to the FDB of the second line card if the record does not already exist in the FDB of the second line card.

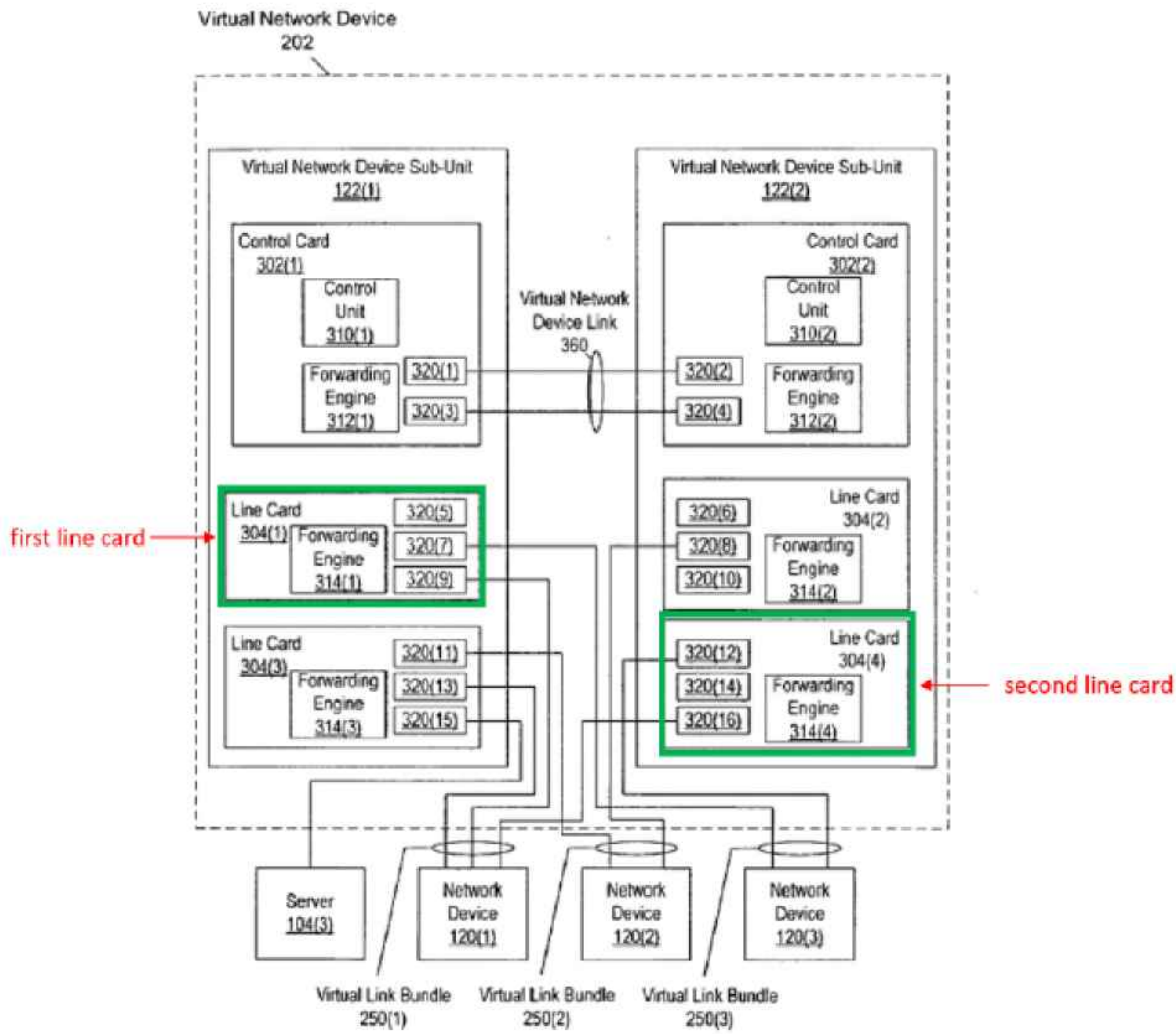


FIG. 3

168. Ishimori's disclosures include:

a. "As a packet forwarding method via a network, there is a method of **learning** a source MAC address of a receive IP packet together with a receive path thereof and, when an IP packet whose destination address is the same MAC address is received, using the corresponding learning result to decide a transmission path thereof. In this situation, when forwarding, at a node that relays IP packets, an IP packet having an unlearned MAC address as a destination

address, a transmission path thereof cannot be specified. As such, in this situation, a broadcast transmission to all nodes (that is, "flooding") is performed. Then, at this time, path **learning** is carried out by **storing** the source MAC address (SA) had by this IP packet and a port that received such **in a buffer had by each node**." Ex. 1005 ¶ 2 (emphasis added).

b. "The path having this representative port leads to card #3 of the nodes 70. At this card #3 as well, no learning result relating to the terminal 20 that is the destination is had. As such, at card #3 as well, this packet is flooded (step P13). In this manner, this packet reaches the terminal 20 that is the destination through card #5. Note that during the packet forwarding from the terminal 10 to the terminal 20 illustrated in FIG. 9 and FIG. 10, at each communication card #0 to #5 of the node groups 60, 70, as described in conjunction with FIGS. 1 to 4, information relating to the path leading to the local device is learned in association with the source address of the packet." *Id.* ¶ 16 (emphasis added).

169. Thus, Ishimori further teaches that the reception path information is learned, which is done by storing the record of association of the MAC address and ingress port to the MAC table. *Id.* ¶ 2. It would have been obvious to do this via the MAC notification as disclosed in Smith. Additionally, during the packet transfer, each of Ishimori's communication cards learns information about the association. Ishimori discloses that "a learn packet is transmitted to all nodes having the corresponding trunk," which includes at least a second line card. *Id.* ¶ 33. Thus, in

Ishimori, each of the communication cards, which includes the second line card, would add a record of the association to the FDB in response to a MAC notification as taught in Smith. The combination of Smith and Ishimori therefore renders obvious this claim.

g. Claim 11[c]: a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards;

170. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 11[c]. Claim 11[c] combines the elements of claims 1[a], 1[b], 1[c], and 11[a]. Therefore, for the same reasons that the combination of Smith, Sharma, and Ishimori discloses claims 1[a], 1[b], 1[c], and 11[a], Smith, Sharma, and Ishimori also disclose claim 11[c]. *See supra* ¶¶ 112–120, 153–155, 146.

h. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data

packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.

171. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 11[d]. The combination of Smith, Sharma, and Ishimori disclose this limitation for the same reasons that the combination of Smith, Sharma, and Ishimori discloses claims 1[d], 1[e], 1[f], 1[g], and 1[h]. *See supra* ¶¶ 121–135, 156–161.

i. Claim 12: The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.

172. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 12. *See supra* ¶¶ 162–163.

j. Claim 13: The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.

173. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 13. *See supra* ¶¶ 164–166.

3. Ground 3: Claims 1, 4–7, 10–11, 14–17, and 20 would have been obvious over the Smith-Sharma-Ishimori-Edsall combination

174. I have analyzed claims 1, 4–7, 10–11, 14–17, and 20 and conclude that they would have been obvious over Smith in view of Sharma and Ishimori in further view of Edsall. I provide a detailed analysis of each claim limitation below.

a. Claim 1[g]: checking said MAC source address of the data packet against records in said FDB of said first line card; and

175. In my opinion, to the extent this is not disclosed by the combination of Smith, Sharma, and Ishimori, it is disclosed in further view of Edsall. Edsall teaches that the forwarding engine learns the source MAC address of a frame received at the ingress card for the first time. "If the frame is received at the ingress card for the first time, the forwarding engine also 'learns' a source MAC address of the frame." Ex. 1006 at 18:39–41. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 18:42–44. "[I]f there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* at 18:54–55.

176. It would have been obvious to use Edsall's learning methods with the Smith-Sharma-Ishimori MAC tables, and there would have been a reasonable expectation of success in doing so.

b. Claim 1[h]: if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record

of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card and said plurality of member line cards.

177. In my opinion, to the extent the combination of Smith, Sharma, and Ishimori does not render obvious this limitation, it would have been obvious to combine Smith, Sharma, and Ishimori with Edsall. Edsall teaches that the forwarding engine learns the source MAC address of a frame received at the ingress card for the first time. *Id.* at 18:39– 41. Edsall further teaches that "if there is not a current entry," the forwarding engine "learn[s] the source address/index of the frame." *Id.* at 18:54–55. It does so by "creating/updating an entry of the L2 forwarding table with the source MAC address and its location (index) within the switch." *Id.* at 18:42–44. It then floods copies of the fabric frame to all the egress line cards of the network switch, which Edsall calls the "flood-to-fabric (FF) operation." *Id.* at 18:47–50. This "forces each forwarding engine associated with each egress card to either (i) update its current L2 forwarding table entry with the newly-learned source MAC address and index of the frame or, if there is not a current entry, (ii) learn the source address/index of the frame." *Id.* at 18:50–55.

178. It would have been obvious to add Edsall's learning method and flood-to-fabric operation in the Smith-Sharma-Ishimori path learning operations. A POSITA would have been able to implement this teaching with a reasonable expectation of success, because the combination of Smith and Ishimori already teach

that a message of new associations are sent to the plurality of member line cards. Ex. 1004 ¶ 63; Ex. 1005 ¶ 16. It would have been within the knowledge of a POSITA to create new entries in a FDB when an association of a MAC address and port does not yet exist. Thus, the combination of Smith, Sharma, Ishimori, and Edsall renders obvious this limitation.

c. Claim 4: The method according to claim 3 and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

179. In my opinion, the combination of Smith, Ishimori, and Edsall discloses claim 4. Edsall discloses that the "PI indicator" is asserted when a forwarding table entry for the MAC address is learned through one of the ports (*e.g.*, data packets transmitted via a port of the line card) "as opposed to through the switch fabric" (*e.g.*, received from another of the line cards). Ex. 1006 at 6:46–64. Thus, the PI indicator is different for the MAC address learned through one of the ports, which correspond to the first type of records in the claim ("data packets transmitted via a port of the line card") "as opposed to through the switch fabric, which corresponds to the second type of records ("messages from another of the line cards"). Ex. 1006 at 6:46–64.

180. Additionally, during prosecution, the Examiner found that Edsall disclosed this limitation and the applicant did not amend the claims based on this rejection. Ex. 1002 at 82, 120–122. The Examiner determined that Edsall also discloses that the "PI indicator is asserted for a destination MAC address entry of the forwarding table on the egress card and the DI contained in the switched fabric frame (*i.e.*, the ingress DI) is different from the DI stored in this egress forwarding table (*i.e.*, the egress DI)." Ex. 1006 at 18:56–19:5; *see also* Ex. 1002 at 82. Based on my review of the file history, it is my understanding that this disclosure from Edsall was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 120–122.

d. Claim 5[a]: The method according to claim 4, and comprising: associating a respective aging time with each of the records;

181. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 5[a]. Ishimori's disclosures include:

a. "Furthermore, in this system, based on an approach of effectively utilizing a limited buffer storage capacity, upon learning, when there is no DA search hit for the same address in a certain period, the learning result relating to this MAC address is deleted. This operation leading to deletion is referred to as "**aging.**" Specifically, as illustrated in FIG. 5, the buffers of the cards of each node have **a "hit bit" corresponding to the learned MAC address; when this is learned, the hit bit = "1"**. Then, at the **first aging process, this**

hit bit is cleared (that is, made to be "0"; steps S11, S12). Here, the learning result is not yet deleted. Then, **when the hit bit is still "0" at the next aging process** (step S14, step S15, "No" at step S11), the **corresponding MAC address is deleted from the buffer** (step S13). Meanwhile, when overwriting learning is performed previous to this or when there is a hit at the DA search (that is, the destination search operation of step S3 in FIG. 2 above), the **hit bit is again set to "1"** (as a result, step S11 "Yes" and step S12). This configuration prevents an address currently being used (that is, being used for communication) from being deleted from the buffer." Ex. 1005 ¶ 9 (emphasis added).

b. "This aging process is further described in conjunction with FIGS. 6 to 8. Steps P1 to P6 in FIG. 7 are the same steps as steps P1 to P6 in FIG. 4 above. Now, each time an IP packet is forwarded, at steps P1, P2, the MAC address "A" is learned at card #0, and the hit bit thereof is made to be "1". Likewise, at steps P2, P3, the MAC address "A" is learned at card #1 and card #2. Continuing in the same vein, at steps P4, P5 and steps P5, P6, the MAC address "B" is learned at card #2 and card #0." *Id.* ¶ 10 (emphasis added).

c. "Afterward, as illustrated in the uppermost row in FIG. 8, the first aging process clears the respective hit bits of the MAC addresses "A", "B" at each card #0, #1, #2 (steps S12). It is then supposed that before the second aging process, the next IP packet, likewise addressed to terminal B 20, is sent from terminal A 10 (step P7). At card #0 that receives this, the source MAC address "A" is learned by overwriting. At the same time, in the destination search, the destination MAC address "B" becomes the search target. As such,

the MAC addresses "A", "B", which are temporarily cleared to "0" as above, have their hit bits respectively returned to "1"." *Id.* ¶ 11 (emphasis added).

d. "Then, this IP packet is forwarded to card #2 according to the learning result regarding the MAC address "B" that yielded a hit in the destination search (step P8). Then, in card #2 as well, theMAC addresses "A", "B" temporarily cleared to "0" as above have their hit bits respectively returned to "1" by the same operation as above. It is then supposed that afterward, the next aging-process timing arrives without either of the IP packets of terminals A, B 10, 20 being received. Here, as illustrated in the lowermost row in FIG. 8, at card #0 and card #2, as above, the hit bits of the MAC addresses "A", "B" are each returned to "1" at steps P7, P8. As such, these are again cleared to "0". However, the corresponding learning result is not yet erased at this point (step S12). Meanwhile, for card #1, no corresponding IP packet passes through during this time, and the hit bits of the MAC addresses "A", "B" remain "0". As such, the corresponding learning result is deleted at this point (step S13)." *Id.* ¶ 12 (emphasis added).

182. Ishimori teaches that "the buffers of the cards of each node have a 'hit bit' corresponding to the learned MAC address; when this is learned, the hit bit = '1'." *Id.* ¶ 9. Therefore, Ishimori associates a respective aging time with each of the records. "[A]t the first aging process, this hit bit is cleared." *Id.* "[W]hen the hit bit is still '0' at the next aging process," "the corresponding MAC address is deleted

from the buffer." *Id.* Thus, the hit bit is how Ishimori associates a respective aging time with each of the records.

183. It would have been obvious to implement a "hit bit" on the lookup tables in the line cards of Smith, Sharma, and Edsall (where Edsall leverages the PI Indicator) in order to associate a respective aging time with each of the records.

e. Claim 5[b]: refreshing the records in the FDB responsively to further packets transmitted by the line cards; and

184. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 5[b]. As described above, Ishimori teaches that when the hit bit is "0," "the corresponding MAC address is deleted from the buffer." *Id.* ¶¶ 9, 11. Ishimori then teaches that when the destination MAC address is received at card #0, the source MAC address is learned by "overwriting," and the hit bit is "returned to '1'" (*e.g.*, refreshing the records in the FDB responsively to further packets transmitted by the line cards). *Id.* ¶ 11.

185. It would have been obvious to a POSITA implement a "hit bit" on the lookup tables in the line cards of Smith, Sharma, and Edsall (where Edsall leverages the PI Indicator) and to refresh the hit bit by learning the source MAC address of the data packets transmitted by the line cards.

f. Claim 5[c]: removing the records from the respective FDB if the records are not refreshed within the respective aging time.

186. In my opinion, the combination of Smith, Sharma, and Ishimori discloses claim 5[c]. Ishimori teaches that if the hit bit remains "0" in the next aging process, then the corresponding MAC address is deleted. *Id.* ¶¶ 9, 12. Thus, Ishimori discloses removing the records from the FDB if the records are not refreshed within the respective aging time.

187. It would have been obvious to a POSITA implement a "hit bit" on the lookup tables in the line cards of Smith, Sharma, and Edsall (where Edsall leverages the PI Indicator) to remove the records from the respective lookup tables in Smith-Sharma-Edsall if the record has not been refreshed by the next aging process as taught by Ishimori.

g. Claim 6: The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via switching core of the network node to at least the second line card.

188. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 6. As I explained above, Smith discloses this limitation. *See supra* ¶¶ 142– 143. Edsall further discloses the "plurality of line cards" are "interconnected by a switch fabric 550." Ex. 1006, 8:20–27. The "switch fabric" in Edsall could be a "switching core." Thus, a POSITA would have understood that the packet is sent from the first line card via a switching core to at least the second line card.

h. Claim 7: The method according to claim 6, wherein sending the synchronization packet comprises, if the

record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.

189. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 7. Smith teaches that its virtual network device includes several line cards, which include several interfaces. Ex. 1004 ¶ 46. "[W]hen updating control protocol behavior of virtual link bundle 250(1), a user can simply access virtual network device sub-unit 122(1) (instead of accessing both virtual network device sub-units 122(1) and 122(2)). *Id.* Virtual network device sub-unit 122(1) can then automatically propagate to network device 122(2) any changes made by the user to the control protocols." *Id.* ¶ 59. Smith teaches that "MAC notification frames are used to keep the content of the L2 tables in virtual network device sub-unit 122(1) synchronized with the content of the L2 tables in virtual network device sub-unit 122(2) and vice versa." *Id.* ¶ 62.

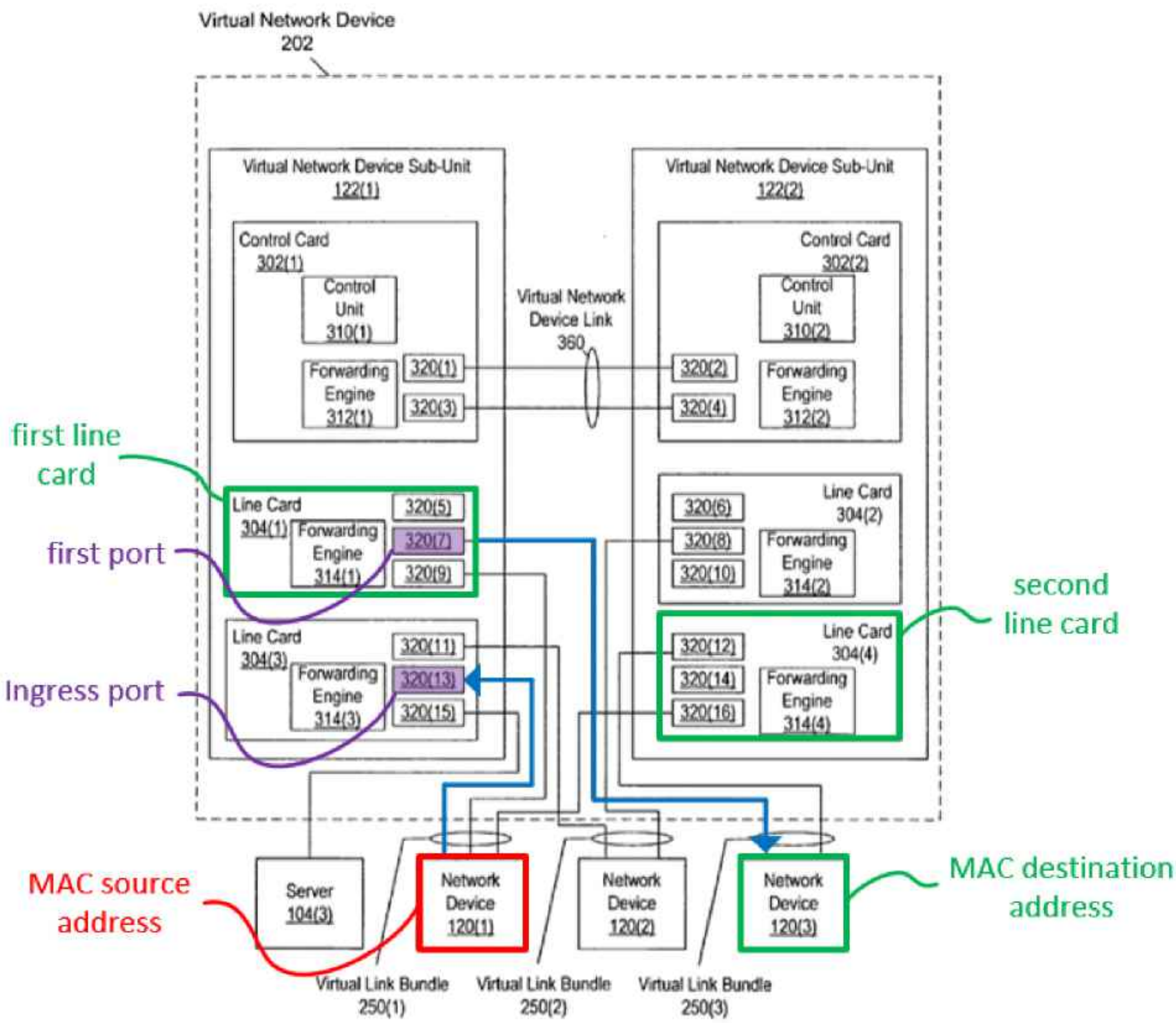


FIG. 3

190. Thus, if the record in the FDB of a line card (e.g., line card 304(3)) associates the MAC source address (e.g., virtual device 120(1)) with a port different from the one of the ports on which the data packet was received (e.g., a port other than 320(13)), the MAC notification frames will notify and update the L2 tables in the virtual network device 202, which would include at least the second line card (e.g., line card 304(4)) to indicate the record has been changed. Similarly, Ishimori

teaches that the routes are transmitted via a "learn packet." Ex. 1005 ¶ 25. A POSITA would understand that the "learn packet" as taught in Ishimori functions as a synchronization packet. During the packet transfer, each of the communication cards learns information about the route (*e.g.*, the second line card). *Id.* ¶¶ 28, 33.

191. Edsall further discloses that the forwarding engine generates an MN frame (*e.g.*, synchronization packet) that may get sent to the SMC (switch management card) to ensure that FwdT0 (*e.g.*, the forwarding table) is synchronized. Ex. 1006 at 17:26–38. The forwarding engine also asserts an appropriate bit of the POE field (port-of-exit field) when generating the MN frame, which is a port different from one of the ports on which data was received. *Id.* It would have been obvious to a POSITA to implement the MN frame from Smith to include the POE field in order to note the port interface of the switch fabric. *Id.* at 6:24–25. This disclosure was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83, 120–122. Thus, the combination of Smith, Sharma, Ishimori, and Edsall discloses this limitation.

i. Claim 10[a]: The method according to claim 1, and comprising: conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;

192. In my opinion, Smith discloses claim 10[a]. *See also supra* ¶¶ 121–124. Smith teaches that the uplink interface receives a data packet with the "sending device's MAC address," which is a MAC source address. Ex. 1004 ¶ 54. Based on

Figure 3, a further data packet received from a further MAC source address (such as network device 120(3)) would be conveyed to the second line card 304(4) for transmission over the network.

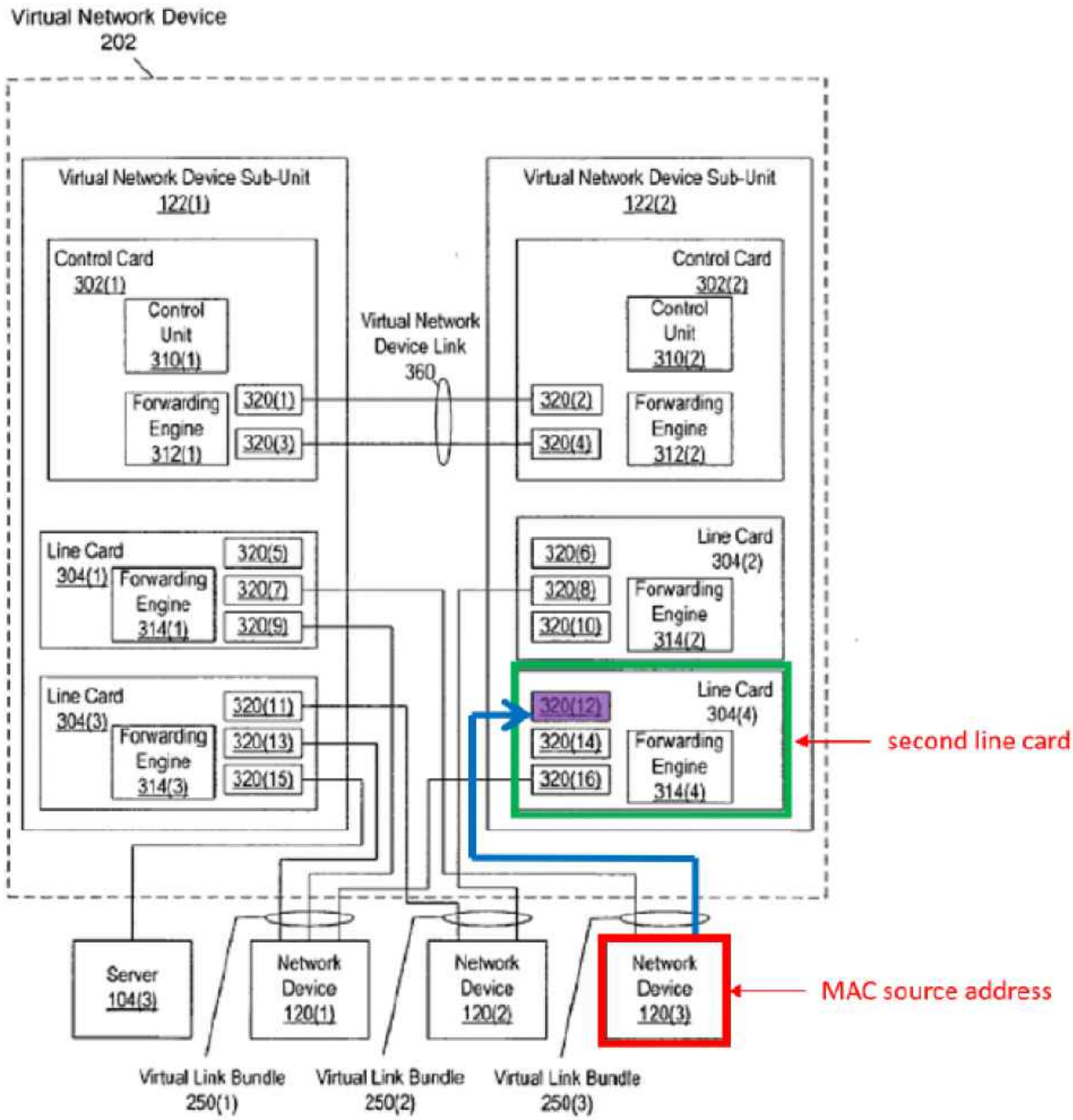


FIG. 3

193. To the extent Smith does not render obvious this claim, Edsall does. Edsall teaches that in each subsequent frame, the encoded address recognition logic (EARL) circuit looks up the MAC address and "sends the corresponding rewrite information over the local bus after the frame" (*e.g.*, conveying a further data packet to the second line card for transmission over the network). Ex. 1006 at 14:22–34. This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122.

j. Claim 10[b]: checking the further MAC source address against the records in the FDB of the second line card; and

194. In my opinion, Smith discloses claim 10[b]. *See also supra* ¶¶ 131–138. Smith teaches that that the virtual network device sub-unit learns the source identifier of the sending device, which is the MAC source address. Ex. 1004 ¶ 65. These identifiers are stored in a lookup table (*e.g.*, FDB) on virtual network device sub-unit. *Id.* ¶ 61. The MAC source address, which is the MAC address of network device 120(3), would be checked in the records of the FDB of said second line card 304(4).

195. To the extent Smith does not disclose this limitation, Edsall does. Edsall teaches checking to see if the rewrite information matches, in other words, checking the MAC source address against the records in the FDB of the second line card). Ex. 1006 at 14:22–34 ("The destination port circuitry (or, alternatively, a

UDlink or central rewrite engine) matches the frame with the rewrite information and modifies the frame as needed by replacing, *inter alia*, the destination and source MAC addresses."). This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122.

k. Claim 10[c] responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.

196. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall renders obvious claim 10[c]. *See also supra* ¶¶ 131–138. Smith teaches that that the network device sub-unit 122(2) sends a MAC notification (*e.g.*, a message) to update the forwarding engines (*e.g.*, sending a further message to inform at least the first line card) when it learns of a new association (*e.g.*, the further record). Ex. 1004 ¶ 63. "After being updated based on the MAC notification, the forwarding engines in the virtual network device sub-unit 122(1) now know the location of the device identified by the destination address." *Id.* A POSITA would understand that the MAC notification (*e.g.*, message) is therefore sent to inform at least the first member line card (*e.g.*, line card 304(1)) of the new association.

197. Edsall further teaches that the forwarding engine "modifies the frame as needed by replacing . . . the destination and source MAC addresses" (*e.g.*, adding

a further record with respect to the MAC source address to the FDB). Ex. 1006 at 14:22–34. This was cited by the examiner during prosecution and the applicant did not amend the claims based on this rejection. Ex. 1002 at 83–84, 120–122. It would have been obvious to use any of the learning methods from Smith, Sharma, Ishimori, or Edsall, which all disclose updating a second line card with a further record and sending a message to inform at least the first line card of the further record.

l. Claim 11[a]: a switching core;

198. To the extent this is not disclosed by the combination of Smith, Sharma, and Ishimori, Edsall discloses this. The '400 patent describes the "switching core" as linking the multiple line cards. Ex. 1001 at 6:8–10. Edsall teaches that the "plurality of line cards . . . are interconnected by a switch fabric 550." Ex. 1006 at 8:20–27. Thus, the "switching fabric" in Edsall could be a "switching core."

m. Claim 11[d]: wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in

said FDB, to flood said data packet via one and only one of said LAG ports.

199. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall renders obvious claim 11[d]. This limitation is a combination of claims 1[d], 1[e], 1[f], 1[g], and 1[h]. Therefore, the combination of Smith, Sharma, Ishimori, Edsall renders obvious this limitation for the same reasons that the combination of Smith, Sharma, Ishimori, and Edsall renders obvious claims 1[d], 1[e], 1[f], 1[g], and 1[h]. *See supra* ¶¶ 121–138, 172–173.

n. Claim 14: The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.

200. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 14. *See supra* ¶¶ 176–177.

o. Claim 15: The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.

201. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 15. *See supra* ¶¶ 178–184.

p. Claim 16: The node according to claim 11, wherein the message comprises a synchronization packet, which is

transmitted from the first line card via the switching core to at least the second line card.

202. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 16. *See supra* ¶ 185.

q. Claim 17: The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.

203. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 17. *See supra* ¶¶ 186–188.

r. Claim 20: The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

204. In my opinion, the combination of Smith, Sharma, Ishimori, and Edsall discloses claim 20. *See supra* ¶¶ 189–194.

4. Ground 4: Claims 8–9 and 18–19 would have been obvious over the Smith-Sharma-Ishimori-Zelig combination

205. I have analyzed claims 8–9 and 18–19 and conclude that they would have been obvious over Smith in view of Sharma and Ishimori in further view of Zelig. I provide a detailed analysis of each claim limitation below.

a. Claim 8: The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

206. In my opinion, the combination of Smith, Sharma, Ishimori, and Zelig discloses claim 8. Zelig's disclosures include:

a. "In another preferred embodiment, at least one of the backup virtual bridges is connected by a single secondary virtual connection to a selected one of the primary virtual bridges, such that upon the failure of the at least one of the primary virtual bridges with which the at least one of the backup virtual bridges is associated, the at least one of the backup virtual bridges transmits and receives the data packets over the network via the selected one of the primary virtual bridges over the single secondary virtual connection. Preferably, **each of the primary virtual bridges** is adapted to **maintain a respective media access control (MAC) table**, and to **forward the data packets in accordance with entries in the MAC table**, and wherein each of the primary virtual bridges is adapted, upon detecting the failure of the at least one of the primary virtual bridges, to update the entries in the respective MAC table that point to the at

least one of the primary virtual bridges so as to point instead to the selected one of the primary virtual bridges." Ex. 1007 ¶ 31 (emphasis added).

b. "FIG. 1 is a block diagram that schematically illustrates a **VPN** 20 with a hierarchical **VPLS** topology, implementing a protection Scheme in accordance with a preferred embodiment of the present invention. **VPN is built around a virtual private LAN service (VPLS)**, operating within a network 22, typically an IP or MPLS network. **The VPLS is based on virtual bridges** 30, 32,34, 36,38, and 40,or VPLS-capable PEs, which are connected by PWs 70, 72, 74 and 76 through network 22. Although for clarity of illustration, network 22 includes only a small number of PES and represents only a single VPLS instance, the principles embodied in this network may be extended in a straightforward manner to larger networks and to multiple VPLS instances." *Id.* ¶ 42 (emphasis added).

c. "Three primary virtual bridges 30, 32 and 34, referred to as primary core nodes, are connected with each other in a full mesh with PW connections 70. Typically, the PW connections comprise MPLS tunnels, but they may alternatively comprise virtual connections of other types, such as GRE or L2TP tunnels. Each of the primary core nodes 30, 32 and 34, is paired with a corresponding backup virtual bridge, referred to as a standby core node, 36, 38 and 40, respectively. The standby core nodes are connected in the network by redundant backup connections 72. Each standby core node has a topology identical to its corresponding primary core node. For example, standby core node 36 has the same topology image as primary core node 30. Each of the primary and standby core nodes

is connected to all the other core nodes in the network except for the standby or primary core node with which it is paired. An optional connection 79 between a primary core node and its corresponding standby core node may also be included, as described hereinbelow." *Id.* ¶ 43.

207. Zelig discloses that its VPN 20 contains multiple primary virtual bridges 30, 32, and 34. *Id.* ¶¶ 42–43. It would have been obvious to a POSITA that the virtual bridges servicing VPN 20 are configured to serve that VPN. Zelig further teaches that "each of the primary virtual bridges is adapted to maintain a respective media access control (MAC) table, and to forward the data packets in accordance with entries in the MAC table . . . ." *Id.* ¶ 31. Thus, Zelig discloses that each MAC bridge maintains its own MAC table, and because the MAC bridge serves only that VPN instance, the records associating the MAC addresses with the respective ports are maintained independently for each VPN instance.

b. Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

208. In my opinion, the combination of Smith, Sharma, Ishimori, and Zelig discloses claim 9. Smith teaches that the MAC notification is "distributed to any other forwarding engines within virtual network device sub-unit 122(2)." Ex. 1004

¶ 63. Zelig provides that "VPN 20 is built around a virtual private LAN service (VPLS), operating within a network 22." Ex. 1007 ¶ 42.

209. It would have been obvious to a POSITA to adopt the notification scheme in Smith to the architecture in Zelig so that the MAC notification, which is a message, is sent to all the line cards that serve the VPLS instances. Furthermore, the IEEE 802.1Q standards, which is cited by Zelig, Ex. 1007 ¶ 2, disclose a VLAN identifier (VID) as a twelve-bit field that "uniquely identif[ies] the VLAN to which the frame belongs." Ex. 1008 § 9.3.2.3; *see also* Ex. 1016 § 9.6. Thus, it would have been within the knowledge of a POSITA to identify the VPLS instance in the message, such as using the VID, in order to inform all the line cards in the VPLS.

c. Claim 18: The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.

210. In my opinion, the combination of Smith, Sharma, Ishimori, and Zelig discloses claim 18. *See supra* ¶¶ 203–204.

d. Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

211. In my opinion, the combination of Smith, Sharma, Ishimori, and Zelig discloses claim 19. *See supra* ¶¶ 205–206.

5. Ground 5: Claims 9 and 19 would have been obvious over the Smith-Sharma-Ishimori-Zelig-802.1Q combination

212. I have analyzed claims 9 and 19 and conclude that they would have been obvious over the combination of Smith, Sharma, Ishimori, Zelig, and 802.1Q-1998. I provide a detailed analysis of each claim limitation below.

a. Claim 9: The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.

213. In my opinion, the combination of Smith, Sharma, Ishimori, Zelig, and 802.1Q1998 discloses claim 9. *See supra* ¶¶ 205–206.

b. Claim 19: The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.

214. In my opinion, the combination of Smith, Sharma, Ishimori, Zelig, and 802.1Q 1998 discloses claim 19. *See supra* ¶ 208.

IX. ADDITIONAL REMARKS

215. I currently hold the opinions expressed in this Declaration. I reserve the right to further explain and supplement my opinions as I may acquire additional

information and/or attain supplemental insights that may result in added observations.

216. I hereby declare that to the best of my knowledge all statements made are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of this proceeding.

Executed this 3rd day of April, 2023.

DocuSigned by:
Tal Lavian
86EA69B7A94C472...

Dr. Tal Lavian

Appendix A

Tal Lavian, Ph.D.

https://TelecommNet.com
tlavian@TelecommNet.com

Encino, CA 91316
(408)-209-9112

Telecommunications, Network Communications, Mobile Wireless, and Internet Technologies Expert

Dr. Lavian is a scientist, educator, and technologist with over 35 years of experience. He has co-authored over 25 scientific publications, journal articles, and peer-reviewed papers. Dr. Lavian is an expert in network communications and telecommunications, including Internet protocols, data communications, and computer networks.

Dr. Lavian has spent 20 years researching, studying, and lecturing at UC Berkeley's College of Engineering. His research focuses on telecommunications systems, data networks, network services, software, network protocols, and communications frameworks. He holds a Ph.D. in Computer Science from UC Berkeley (2006), specializing in network communications; M.Sc., Electrical Engineering ('97) from Tel Aviv University; and B.Sc., Mathematics and Computer Science ('87)

EXPERTISE

Network communications, telecommunications, Internet protocols, and mobile wireless:
• **Network Communications:** Internet protocols; TCP/IP suite, TCP, UDP, IP, Ethernet, 802.3, network protocols, network software applications, data link, network, transport layers, SNMP, NMS, network management, packet switching, and network architecture.
• **VoIP/Streaming Media:** VoIP, SIP, RTP, video/audio conferencing, streaming media, IP telephony, transport systems, PSTN, circuit switching, WebRTC, SS7, SONET, and TDM.
• **Mobile wireless:** Wi-Fi, 802.11, Bluetooth, Wireless LAN (WLAN), MAC, PHY, ARQ, HARQ. Cellular, SMS, MMS, instant messaging (chat), mobile devices, and smartphones.
• **Internet/cloud:** Internet Technologies, Web applications, HTTP, e-mail, SMTP, POP, IMAP, firewalls, security, FTP, client-server, cloud computing, and distributed computing.
• **Routing/switching:** LAN, WAN, VPN, encapsulation, routing protocols, RIP, BGP, MPLS, OSPF, multicast, VPLS, Pseudowire, DNS, QoS, queuing, traffic control, network infrastructure, and architectures.

Dr. Lavian has extensive experience in the software development of computer networks, architectures, configurations, installations, and network testing. He has academic and hands-on experience in the above fields, including technology products from different companies, implementations, related standards, designs, systems, hardware, and software technologies.

ACCOMPLISHMENTS

- Principal Investigator (PI) for three US Department of Defense (DARPA) projects.
  - Directed networking computation project for the US Air Force Research Lab (AFRL).
  - PI of a wireless research project for an undisclosed US federal agency.
- An inventor of over 120 patents, over 60 prosecuted *pro-se* before the USPTO.
- Led and developed the first network resource scheduling service for grid computing.

- Managed and engineered the first demonstrated dynamic transatlantic allocation of 10Gbs Lambdas as a grid service.
- Development and successfully demonstrated the first wire-speed active network on commercial hardware.
- Created and chaired Nortel Networks' EDN Patent Committee.

PROFESSIONAL EXPERIENCE

The University of California, Berkeley, Berkeley, California 2000-2019

U.C. Berkeley SkyDeck, Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, Nortel's Scientist Liaison

Some positions and projects were concurrent, others sequential.

- U.C. Berkeley SkyDeck startups - advanced technology research, development, business, and market.
- Industry fellow and lecturer at the Sutardja Center for Entrepreneurship and Technology (SCET).
- Conducted research projects in data centers (RAD Labs), telecommunication infrastructure (SAHARA), and wireless systems (ICEBERG).
- Acted as a scientific liaison between Nortel Research Lab and U.C. Berkeley, providing tangible value in advanced technologies.
- Developed long-term technology for the enterprise market, integrating communication and computing technologies.
- Studied network services, telecommunication systems and software, communications infrastructure, and data centers.
- Earned a Ph.D. in Computer Science with a specialization in network communications.

TelecommNet Engineering, Inc. Sunnyvale, California 2006-Present

Principal Scientist

- Consulting in network communications, telecommunications, Internet protocols, and smartphone mobile wireless devices.
- Providing system architecture and technology analysis for computer networks, mobile wireless devices, and Internet web technologies projects.
- Providing expert witness services in network communications patent infringement lawsuits.

CRadar.Ai, U.C. Berkeley, California 2018-2019

CTO / Principal Investigator

- CRadar.Ai improves the Radar wireless RF signal phase noise purity by 100x.
- Accurate Radars are paramount for self-driving car safety. Radars "see" where Cameras and LiDars are "blind" (fog, rain, snow, direct sunlight, and darkness).
- The superior wireless RF signal quality provides a clean signal for high Radar accuracy.
- Improving Radar accuracy and resolution enables genuine redundancy and sensory fusion and puts the Radar into the sensory spearhead.

Aybell (VisuMenu Inc.), U.C. Berkeley, California 2016-Present

CEO/CTO

- Aybell transforms smartphones into visual menu systems, making the phone a frictionless point for user interactions with customer service platform features. Empowers consumers to reach suitable agents in call centers, overcoming customer service barriers. Aybell is a branding and marketing of VisuMenu advanced technologies.
- Architecture, design, and implementation of a cloud data center for connecting smartphone users to any company and service by digitizing interactive voice systems and exposing APIs to other applications through cloud service.
- The system was deployed as a cloud networking and cloud computing service on Amazon Web Services (AWS) and Google Cloud Platform (GCP).
- Technologies include Data Science analytics, Machine Learning (ML), Artificial Intelligence (AI), and Statistical Learning (SL). Building an NLP Parser using Python, NLTK, SpaCy, and other NLP libraries and modules.

VisuMenu, Inc., Sunnyvale, California 2010-2016

Co-Founder and Chief Technology Officer (CTO)

- Led the software design and development of a visual IVR system for smartphones and other mobile devices, based on the innovative use of wireless and network communications technologies.
- Designed a voice search engine for IVR / PBX using Asterisk, SIP, and VoIP.
- The system was deployed as a cloud networking and cloud computing service on Amazon Web Services (AWS) and Google Cloud Platform (GCP).
- VisuMenu advanced technologies rebranded as Aybell.

Ixia, Santa Clara, California 2008 - 2008

Network Communications Consultant

Researched and developed advanced network communications testing technologies:

- IxNetwork/IxN2X —IP routing, switching devices, and broadband access equipment. Provided traffic generation and emulation for the full range of protocols: OSPF, RIP, EIGRP, BGP, IS-IS, MPLS, unicast, multicast, broadcast, layer 2/3 VPNs, IPSec, carrier Ethernet, broadband access, and data center bridging. Tested and validated IEEE, ITU, and IETF RFC standards compatibility.
- IxLoad — quickly and accurately modeled high-volume video, data, and voice subscribers and servers to test the real-world performance of multiservice delivery and security platforms.
- IxCatapult — emulated a broad range of wireless access and core protocols to test wireless components and systems that, combined with IxLoad, provide an end-to-end solution for testing wireless service quality.
- IxVeriWave — employed a client-centric model to test Wi-Fi and wireless LAN networks by generating repeatable large-scale, real-world test scenarios that are virtually impossible to create by any other means.
- Test automation — provided simple, comprehensive lab automation to help test engineering teams create, organize, catalog, and schedule execution of tests.

Nortel Networks, Santa Clara, California 1996 - 2007

Employed initially by Bay Networks, later acquired by Nortel Networks

Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer
Held scientific and research roles at Nortel Labs, Bay Architecture Labs, and the CTO's office.

Principal Investigator for U.S. Department of Defense (DARPA) Projects
- Conceived, proposed and completed three research projects: active networks, DWDM-RAM, and a networking computation project for Air Force Research Lab (AFRL).
- Led a wireless research project for an undisclosed U.S. federal agency.

Academic and Industrial Researcher
- Analyzed new technologies to reduce risks associated with R&D investment.
- Headed research collaboration with leading universities and professors at U.C. Berkeley, Northwestern University, University of Amsterdam, and the University of Technology, Sydney.
- Evaluated competitive products relative to Nortel's products and technology.
- Proactively identified prospective business ideas, leading to new networking products.
- Predicted technological trends through researching the technological horizon and academic sphere.
- Designed software for switches, routers, and network communications devices.
- Developed systems and architectures for switches, routers, and network management.
- Researched and developed the following projects:
  - Data-Center Communications: network and server orchestration 2006-2007
  - DRAC: SOA-facilitated L1/L2/L3 network dynamic controller 2003-2007
  - Omega: classified project for undisclosed U.S. Federal Agency 2006-2006
  - Platform project for the U.S. Air Force Research Laboratory (AFRL) 2005-2005
  - Network resource orchestration for Web services workflows 2004-2005
  - A proxy study between Web/grids services and network services 2004-2004
  - Streaming content replication: real-time A/V media multicast at edge 2003-2004
  - DWDM-RAM: U.S. DARPA-funded program on agile optical transport 2003-2004
  - Packet capturing and forwarding service on IP and Ethernet traffic 2002-2003
  - CO2: content-aware agile networking 2001-2003
  - Active networks: US DARPA-funded research program 1999-2002
  - ORE: programmable network service platform 1998-2002
  - JVM platform: Java on network devices 1998-2001
  - Web-based device management: network device management 1996-1997

Technology Innovator and Patent Leader
- Created and chaired Nortel Networks' EDN Patent Committee.
- Facilitated a continuous stream of innovative ideas and their conversion into intellectual property rights.
- Developed intellectual property assets through invention and analysis of existing technology portfolios.

<u>Aptel Communications</u>, Netanya, Israel 1994-1995

Software Engineer, Team Leader
Start-up company focused on mobile wireless CDMA spread spectrum PCN/PCS.
- Developed a mobile wireless device using an unlicensed band - Direct Sequence Spread Spectrum (DSSS); FCC part 15 - unlicensed transmitters.
- Designed and managed a personal communication network (PCN) and personal communication system (PCS), which were the precursors of short text messages (SMS).
- Designed and developed network communications software products in C/C++.
- Invented and implemented a two-way paging product.

Scitex Ltd., Herzeliya, Israel 1990-1993
Software Engineer, Team Leader
Software and hardware company acquired by Hewlett Packard (HP)
- Developed system and network communications in C/C++.
- I provided IT services, System Administration, and network administration.
- I worked on Unix systems, including IBM AIX, HP, and SUN Unix.
- Invented Parallel SIMD Architecture.
- Participated in the Technology Innovation group.

Shalev, Ramat-HaSharon, Israel 1987-1990
Start-up company
Software Engineer
- Developed real-time software and algorithms in C/C++ and Pascal.

PROFESSIONAL ASSOCIATIONS

- IEEE senior member
- IEEE CNSV co-chair, Intellectual Property SIG (2013)
- President Next Step Toastmasters (an advanced TM club in the Silicon Valley) (2013-2014)
- Technical co-chair, IEEE Hot Interconnects 2005 at Stanford University
- Member, IEEE Communications Society (COMMSOC)
- Member, IEEE Computer Society
- Member, IEEE Systems, Man, and Cybernetics Society
- Member, IEEE-USA Intellectual Property Committee (2012)
- Member, ACM, ACM Special Interest Group on Data Communication (SIGCOM)
- Member, ACM Special Interest Group on Hypertext, Hypermedia, and Web (SIGWEB)
- Member, IEEE Consultants' Network (CNSV)
- Global Member, Internet Society (ISOC)
- President Java Users Group – Silicon Valley Mountain View, CA,1999-2000
- Toastmasters International

FORMER ADVISORY BOARDS POSITIONS

- Quixey –search engine for wireless mobile apps
- Mytopia – mobile wireless social games
- iLeverage – Israeli Innovations

PROFESSIONAL AWARDS

- Top Talent Award – Nortel
- Top Inventors Award – Nortel EDN
- Certified IEEE-WCET - Wireless Communications Engineering Technologies (2012)
- Toastmasters International - Competent Communicator (twice)
- Toastmasters International - Advanced Communicator Bronze
- Best Paper Presentation Award - ICE/IEEE Conference. "R&D Models for Advanced Development & Corporate Research"

PERSONAL

- USA FIT – San Jose Marathon running club (2017-2020)

Patents and Publications

Patents Issued
(Representative List)

US 9,690,877	Systems and methods for electronic communications	Link
US 9,660,655	Ultra-low phase noise frequency synthesizer	Link
US 9,184,989	Grid proxy architecture for network resources	Link
US 9,521,255	Systems and methods for visual presentation and selection of IVR menu	Link
US 9,083,728	Systems and methods to support sharing and exchanging in a network	Link
US 9,021,130	Photonic line sharing for high-speed routers	Link
US 8,762,963	Translation of programming code	Link
US 8,762,962	Methods and apparatus for automatic translation of a computer program language code	Link
US 8,745,573	Platform-independent application development framework	Link
US 8,731,148	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,688,796	Rating system for determining whether to accept or reject an objection raised by a user in a social network	Link
US 8,619,793	Dynamic assignment of traffic classes to a priority queue in a packet-forwarding device	Link
US 8,572,303	A portable Universal communication device	Link
US 8,553,859	Device and method for providing enhanced telephony	Link
US 8,548,131	Systems and methods for communicating with an interactive voice response system	Link
US 8,537,989	Device and method for providing enhanced telephony	Link
US 8,341,257	Grid proxy architecture for network resources	Link
US 8,161,139	Method and apparatus for intelligent management of a network element	Link
US 8,146,090	Time-value curves to provide dynamic QoS for time-sensitive file transfer	Link
US 8,078,708	Grid proxy architecture for network resources	Link
US 7,944,827	Content-aware dynamic network resource allocation	Link

US 7,860,999	Distributed computation in network devices	Link
US 7,734,748	Method and apparatus for intelligent management of a network element	Link
US 7,710,871	Dynamic assignment of traffic classes to a priority queue in a packet-forwarding device	Link
US 7,580,349	Content-aware dynamic network resource allocation	Link
US 7,433,941	Method and apparatus for accessing network information on a network device	Link
US 7,359,993	Method and apparatus for external interfacing resources with a network element	Link
US 7,313,608	Method and apparatus for using documents written in a markup language to access and configure network elements	Link
US 7,260,621	The object-oriented network management interface	Link
US 7,237,012	Method and apparatus for classifying Java remote method invocation transport traffic	Link
US 7,127,526	Method and apparatus for dynamically loading and managing software services on a network device	Link
US 7,047,536	Method and apparatus for classifying remote procedure call transport traffic	Link
US 7,039,724	Programmable command-line interface API for managing the operation of a network device	Link
US 6,976,054	Method and system for accessing low-level resources in a network device	Link
US 6,970,943	Routing architecture includes a compute plane configured for high-speed processing of packets to provide application layer support.	Link
US 6,950,932	Security association mediator for Java-enabled devices	Link
US 6,850,989	Method and apparatus for automatically configuring a network switch	Link
US 6,845,397	Interface method and system for accessing inner layers of a network protocol	Link
US 6,842,781	Download and processing of a network management application on a network device	Link
US 6,772,205	Executing applications on a target network device using a proxy network device	Link
US 6,564,325	Method of and apparatus for providing multi-level security access to a system	Link
US 6,175,868	Method and apparatus for automatically configuring a network switch	Link
US 6,170,015	Network apparatus with Java co-processor	Link
US 8,687,777	Systems and methods for visual presentation and selection of IVR menu	Link

US 8,681,951	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,625,756	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,594,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,548,135	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,406,388	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,345,835	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,223,931	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,160,215	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,155,280	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,054,952	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,000,454	Systems and methods for visual presentation and selection of IVR menu	Link
EP 1,905,211	A technique for authenticating network users	Link
EP 1,142,213	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 9,001,819	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,949,846	Time-value curves to provide dynamic QoS for time-sensitive file transfers	Link
US 8,929,517	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,903,073	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,898,274	Grid proxy architecture for network resources	Link
US 8,880,120	Device and method for providing enhanced telephony	Link
US 8,879,703	System method and device for providing tailored services when a call is on-hold	Link
US 8,879,698	Device and method for providing enhanced telephony	Link
US 8,867,708	Systems and methods for visual presentation and selection of IVR menu	Link
US 8,787,536	Systems and methods for communicating with an interactive voice response system	Link
US 8,782,230	Method and apparatus for using a command design pattern to access and configure network elements	Link
CA 2,358,525	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link

CA 2,989,752	Ultra-low Phase Noise Frequency Synthesizer	Link
US 10,598,764	Radar target detection and imaging system for autonomous vehicles with ultra-low phase noise frequency synthesizer	Link
US 10,404,261	Radar target detection system for autonomous vehicles with an ultra-low phase-noise frequency synthesizer	Link
US 10,348,313	Radar target detection system for autonomous vehicles with an ultra-low phase-noise frequency synthesizer	Link
US 10,205,457	RADAR target detection system for autonomous vehicles with an ultra-low phase-noise frequency synthesizer	Link
US 10,764,264	Technique for authenticating network users	Link
EP 3,311,493	An ultra-low phase-noise frequency synthesizer	Link
US 9,831,881	Radar target detection system for autonomous vehicles with ultra-low phase noise frequency synthesizer	Link
US 9,762,251	Ultra-low phase noise frequency synthesizer	Link
US 9,705,511	Ultra-low phase noise frequency synthesizer	Link

Patent Applications Published and Pending
(Representative List)

US 20150058490	Grid Proxy Architecture for Network Resources	Link
US 20150010136	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20140379784	Method and Apparatus for Using a Command Design Pattern to Access and Configure Network Elements	Link
US 20140105025	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140105012	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20140012991	Grid Proxy Architecture for Network Resources	Link
US 20130080898	Systems and Methods for Electronic Communications	Link
US 20130022191	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022183	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20130022181	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20120180059	Time-Value Curves to Provide Dynamic QoS for Time Sensitive File Transfers	Link
US 20120063574	Systems and Methods for Visual Presentation and Selection of IVR Menu	Link
US 20110225330	Portable Universal Communication Device	Link
US 20100220616	Optimizing Network Connections	Link
US 20100217854	Method and Apparatus for Intelligent Management of a Network Element	Link
US 20100146492	Translation of Programming Code	Link
US 20100146112	Efficient Communication Techniques	Link
US 20100146111	Efficient Communication in a Network	Link
US 20090313613	Methods and Apparatus for Automatic Translation of a Computer Program Language Code	Link
US 20090313004	Platform-Independent Application Development Framework	Link
US 20090279562	Content-aware dynamic network resource allocation	Link
US 20080040630	Time-Value Curves to Provide Dynamic QoS for Time Sensitive File	Link

	Transfers	
US 20070169171	A technique for authenticating network users	Link
US 20060123481	Method and apparatus for network immunization	Link
US 20060075042	Extensible Resource Messaging Between User Applications and Network Elements in a Communication Network	Link
US 20050083960	Method and Apparatus for Transporting Parcels of Data Using Network Elements with Network Element Storage	Link
US 20050076339	Method and Apparatus for Automated Negotiation for Resources on a Switched Underlay Network	Link
US 20050076336	Method and Apparatus for Scheduling Resources on a Switched Underlay Network	Link
US 20050076173	Method And Apparatus for Preconditioning Data to Be Transferred on a Switched Underlay Network	Link
US 20050076099	Method and Apparatus for Live Streaming Media Replication in a Communication Network	Link
US 20050074529	Method and apparatus for transporting visualization information on a switched underlay network	Link
US 20040076161	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
US 20020021701	Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device	Link
WO 2006/063052	Method and apparatus for network immunization	Link
WO 2007/008976	A technique for authenticating network users	Link
WO2000/0054460	Method and apparatus for accessing network information on a network device	Link
WO/2016/203460	Ultra-low phase noise frequency synthesizer	Link
WO/2005/033899	Method and apparatus for scheduling resources on a switched underlay network	Link
WO/2000/041368	Dynamic assignment of traffic classes to a priority queue in a packet forwarding device	Link
US 20140156556	A Time-variant rating system and method thereof	Link
US 20140156758	A Reliable rating system and method thereof	Link

US 20170085708	Systems and methods for visual presentation and selection of IVR menu	Link
US 20160373117	Ultra-low phase noise frequency synthesizer	Link
US 20170322687	Systems and methods for electronic communications	Link
US 20170302282	Radar target detection system for autonomous vehicles with ultra-low phase noise frequency synthesizer	Link
US 20180019755	Radar target detection system for autonomous vehicles with ultra-low phase noise frequency synthesizer	Link
US 20170289332	Systems and methods for visual presentation and selection of IVR menu	Link
US 20170269797	Systems and methods for electronic communication	Link
US 20170099058	Ultra-low phase noise frequency synthesizer	Link
US 20170099057	Ultra-low phase noise frequency synthesizer	Link
US 20190128998	Radar target detection and imaging system for autonomous vehicles with ultra-low phase noise frequency synthesizer	Link
US 20190082043	Systems and methods for visual presentation and selection of IVR menu	Link
US 20180146090	Systems and methods for visual presentation and selection of IVR menu	Link
US 20180130102	Reliable rating system and method thereof	Link

Publications
(Representative List)

- "R&D Models for Advanced Development & Corporate Research" Understanding Six Models of Advanced R&D - Ikhlaq Sidhu, Tal Lavian, Victoria Howell - University of California, Berkeley. ASEE Annual Conference and Exposition- 2015. Received "Best Paper Presentation Award" ICE/IEEE Conference June 2015.
- "Communications Architecture in Support of Grid Computing," Tal Lavian, Scholar's Press 2013ISBN 978-3-639-51098-0.
- "Applications Drive Secure Light-path Creation across Heterogeneous Domains, Feature Topic Optical Control Planes for Grid Networks: Opportunities, Challenges, and the Vision." Gommans L.; Van Oudenaarde B.; Dijkstra F.; De Laat C.; Lavian T.; Monga I.; Taal A.; Travostino F.; Wan A.; IEEE Communications Magazine, vol. 44, no. 3, March 2006, pp. 100-106.
- *Lambda Data Grid: Communications Architecture in Support of Grid Computing*. Tal I. Lavian, Randy H. Katz; Doctoral Thesis, University of California at Berkeley. January 2006.
- "Information Switching Networks." Hoang D.B.; T. Lavian; The 4th Workshop on the Internet, *Telecommunications and Signal Processing, WITSP2005*, December 19-21, 2005, Sunshine Coast, Australia.
- "Impact of Grid Computing on Network Operators and HW Vendors." Allcock B.; Arnaud B.; Lavian T.; Papadopoulos P.B.; Hasan M.Z.; Kaplow W.; *IEEE Hot Interconnects at Stanford University 2005*, pp.89-90.
- *DWDM-RAM: A Data Intensive Grid Service Architecture Enabled by Dynamic Optical Networks*. Lavian T.; Mambretti J.; Cutrell D.; Cohen H.J; Merrill S.; Durairaj R.; Daspit P.; Monga I.; Naiksatam S.; Figueira S.; Gutierrez D.; Hoang D.B., Travostino F.; *CCGRID 2004*, pp. 762-764.
- *DWDM-RAM: An Architecture for Data-Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Hoang D.B.; Cohen H.; Cutrell D.; Figueira S.; Lavian T.; Mambretti J.; Monga I.; Naiksatam S.; Travostino F.; Proceedings IEEE Globecom 2004, Workshop on High-Performance Global Grid Networks, Houston, 29 Nov. to 3 Dec. 2004, pp.400-409.
- *Implementation of a Quality of Service Feedback Control Loop on Programmable Routers*. Nguyen C.; Hoang D.B.; Zhao, I.L.; Lavian, T.; Proceedings, 12th IEEE International Conference on Networks 2004. (ICON 2004) Singapore, Volume 2, 16-19 Nov. 2004, pp.578-582.
- *A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks*. Lavian T.; Hoang D.B.; Mambretti J.; Figueira S.; Naiksatam S.; Kaushil N.; Monga I.; Durairaj R.; Cutrell D.; Merrill S.; Cohen H.; Daspit P.; Travostino F; GridNets 2004, San Jose, CA., October 2004.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Daspit, P.; Gutierrez D.; Hoang D. B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F; Proceedings, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004, pp. 707-714.

- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Gutierrez D.; Hoang D.B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004.
- *An Extensible, Programmable, Commercial-Grade Platform for Internet Service Architecture.* Lavian T.; Hoang D.B.; Travostino F.; Wang P.Y.; Subramanian S.; Monga I.; IEEE Transactions on Systems, Man, and Cybernetics on Technologies Promoting Computational Intelligence, Openness and Programmability in Networks and Internet Services Volume 34, Issue 1, Feb. 2004, pp.58-68.
- *DWDM-RAM: An Architecture for Data-Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Lavian T.; Cutrell D.; Mambretti J.; Weinberger J.; Gutierrez D.; Naiksatam S.; Figueira S.; Hoang D. B.; Supercomputing Conference, SC2003 Igniting Innovation, Phoenix, November 2003.
- *Edge Device Multi-Unicasting for Video Streaming.* Lavian T.; Wang P.; Durairaj R.; Hoang D.; Travostino F.; Telecommunications, 2003. ICT 2003. 10th International Conference on Telecommunications, Tahiti, Volume 2, 23 Feb.-1 March 2003 pp. 1441-1447.
- The SAHARA Model for Service Composition Across Multiple Providers. Raman B.; Agarwal S.; Chen Y.; Caesar M.; Cui W.; Lai K.; Lavian T.; Machiraju S.; Mao Z. M.; Porter G.; Roscoe T.; Subramanian L.; Suzuki T.; Zhuang S.; Joseph A. D.; Katz Y.H.; Stoica I.; Proceedings of the First International Conference on Pervasive Computing. ACM Pervasive 2002, pp. 1-14.
- *Enabling Active Flow Manipulation in Silicon-Based Network Forwarding Engines.* Lavian T.; Wang P.; Travostino F.; Subramanian S.; Duraraj R.; Hoang D.B.; Sethaput V.; Culler D.; Proceeding of the Active Networks Conference and Exposition, 2002. (DANCE) 29-30 May 2002, pp. 65-76.
- *Practical Active Network Services within Content-Aware Gateways.* Subramanian S.; Wang P.; Durairaj R.; Rasimas J.; Travostino F.; Lavian T.; Hoang D.B.; Proceeding of the DARPA Active Networks Conference and Exposition, 2002. (DANCE) 29-30 May 2002, pp. 344-354.
- *Active Networking on a Programmable Network Platform.* Wang P.Y.; Lavian T.; Duncan R.; Jaeger R.; Fourth IEEE Conference on Open Architectures and Network Programming (OPEN ARCH), Anchorage, April 2002.
- *Intelligent Network Services through Active Flow Manipulation.* Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; IEEE Intelligent Networks 2001 Workshop (IN2001), Boston, May 2001.
- *Intelligent Network Services through Active Flow Manipulation.* Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; Intelligent Network Workshop, 2001 IEEE 6-9 May 2001, pp.73 -82.
- *Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engine.* Lavian, T.; Wang, P.; Travostino, F.; Subramanian S.; Hoang D.B.; Sethaput V.; Culler D.; Journal of Communications and Networks, March 2001, pp.78-87.
- *Active Networking on a Programmable Networking Platform.* Lavian T.; Wang P.Y.; IEEE Open Architectures and Network Programming, 2001, pp. 95-103.

- *Enabling Active Networks Services on a Gigabit Routing Switch.* Wang P.; Jaeger R.; Duncan R.; Lavian T.; Travostino F.; 2nd Workshop on Active Middleware Services, 2000.
- *Dynamic Classification in Silicon-Based Forwarding Engine Environments.* Jaeger R.; Duncan R.; Travostino F.; Lavian T.; Hollingsworth J.; Selected Papers. 10th IEEE Workshop on Metropolitan Area and Local Networks, 1999. 21-24 Nov. 1999, pp.103-109.
- *Open Programmable Architecture for Java-Enabled Network Devices.* Lavian, T.; Jaeger, R. F.; Hollingsworth, J. K.; IEEE Hot Interconnects Stanford University, August 1999, pp. 265-277.
- *Open Java SNMP MIB API*. Rob Duncan, Tal Lavian, Roy Lee, Jason Zhou, Bay Architecture Lab Technical Report TR98-038, December 1998.
- *Java-Based Open Service Interface Architecture*. Lavian T.; Lau S.; BAL TR98-010 Bay Architecture Lab Technical Report, March 1998.
- *Parallel SIMD Architecture for Color Image Processing*. Lavian T. Tel – Aviv University, Tel – Aviv, Israel, November 1995.
- *Grid Network Services, Draft-ggf-ghpn-netservices-1.0.* George Clapp, Tiziana Ferrari, Doan B. Hoang, Gigi Karmous-Edwards, Tal Lavian, Mark J. Leese, Paul Mealor, InderMonga, Volker Sander, Franco Travostino, Global Grid Forum(GGF).
- *Project DRAC: Creating an applications-aware network.* Travostino F.; Keates R.; Lavian T.; Monga I.; Schofield B.; Nortel Technical Journal, February 2005, pp. 23-26.
- *Optical Network Infrastructure for Grid, Draft-ggf-ghpn-opticalnets-1.* Dimitra Simeonidou, Reza Nejabati, Bill St. Arnaud, Micah Beck, Peter Clarke, Doan B. Hoang, David Hutchison, Gigi Karmous-Edwards, Tal Lavian, Jason Leigh, Joe Mambretti, Volker Sander, John Strand, Franco Travostino, Global Grid Forum(GGF) GHPN Standard GFD-I.036 August 2004.
- *Popeye - Using Fine-grained Network Access Control to Support Mobile Users and Protect Intranet Hosts.* Mike Chen, Barbara Hohlt, Tal Lavian, December 2000.
- Open Networking - Better Networking through Programmability, Open Networking - Better Networking through Programmability
- Dangerous Liaisons – Software Combinations as Derivative Works? Determann L.; Berkeley Technology Law Journal. Volume 21, Issue 4, Fall 2006. (Lavian T. contributor to the technical section).

Presentations and Talks
(Not an exhaustive list)

- Lambda Data Grid
- A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks
- Lambda Data Grid: An Agile Optical Platform for Grid Computing and Data-intensive Applications.
- Workflow Integrated Network Resource Orchestration
- DWDM-RAM: DARPA-Sponsored Research for Data-Intensive Service-on-Demand Advanced Optical Networks
- Impact of Grid Computing on Network Operators and HW Vendors
- Web Services and OGSA
- WINER Workflow Integrated Network Resource Orchestration.
- A Grid Proxy Architecture for Network Resources
- Technology & Society
- Abundant Bandwidth and how it affects us?
- Active Content Networking (ACN)
- DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks
- Application-engaged Dynamic Orchestration of Optical Network Resources
- DWDM-RAM: DARPA-Sponsored Research for Data-Intensive Service-on-Demand Advanced Optical Networks
- An Architecture for Data-Intensive Service Enabled by Next Generation Optical Networks
- A Platform for Data-Intensive Services Enabled by Next Generation Dynamic Optical Networks
- A Platform for Data-Intensive Services Enabled by Next Generation Dynamic Optical Networks
- Optical Networks
- Grid Optical Network Service Architecture for Data-Intensive Applications
- Optical Networking & DWDM
- OptiCal Inc.
- OptiCal & LUMOS Networks
- Optical Networking Services
- Optical Networks
- Business Models for Dynamically Provisioned Optical Networks
- Business Model Concepts for Dynamically Provisioned Optical Networks
- Optical Networks Infrastructure
- Research Challenges in agile optical networks
- Services and Applications' infrastructure for agile optical networks
- Impact on Society
- Technology & Society
- TeraGrid Communication and Computation
- Unified Device Management via Java-enabled Network Devices
- Active Network Node in Silicon-Based L3 Gigabit Routing Switch
- Enabling Active Flow Manipulation (AFM) in Silicon-based Network Forwarding Engines
- Enabling Active Flow Manipulation (AFM) in Silicon-based Network Forwarding Engines

- Active Nets Technology Transfer through High-Performance Network Devices
- Enabling Active Networks Services on A Gigabit Routing Switch
- Programmable Network Node: Applications
- Open Innovation via Java-enabled Network Devices
- Practical Considerations for Deploying a Java Active Networking Platform
- Open Programmable Architecture for Java-enabled Network Devices
- Enabling Active Flow Manipulation In Silicon-based Network Forwarding Engines
- Enabling Active Flow Manipulation In Silicon-based Network Forwarding Engines
- Enabling Active Flow Manipulation In Silicon-based Network Forwarding Engines
- DWDM-RAM: DARPA-Sponsored Research for Data-Intensive Service-on-Demand Advanced Optical Networks
- DWDM-RAM: DARPA-Sponsored Research for Data-Intensive Service-on-Demand Advanced Optical Networks
- Open Programmable Architecture for Java-enabled Network Devices
- Open Java-based Intelligent Agent Architecture for Adaptive Networking Devices
- Edge Device Multi-unicasting for Video Streaming
- Intelligent Network Services through Active Flow Manipulation
- Java SNMP Oplet
- Unified Device Management via Java-enabled Network Devices
- Dynamic Classification in a Silicon-Based Forwarding Engine
- Integrating Active Networking and Commercial-Grade Routing Platforms
- Enabling Active Flow Manipulation In Silicon-based Network Forwarding Engines
- Open Distributed Networking Intelligence: A New Java Paradigm
- Open Networking Better Networking Through Programmability
- Open Networking
- Open Programmability
- Active On A Programmable Networking Platform
- Open Networking Networking through Programmability
- Open Programmable Architecture for Java-enabled Network Devices
- Popeye – Fine-grained Network Access Control for Mobile Users
- Integrating Active Networking and Commercial-Grade Routing Platforms
- Active Networking
- Programmable Network Devices
- Open Programmable Architecture for Java-enabled Network Devices
- To be smart or not to be?

Appendix B

APPENDIX B: CHALLENGED CLAIM LISTING

No.	Limitation
1[pre]	A method for communication, comprising:
1[a]	configuring a network node having a plurality of ports, and at least first and second line cards with respective first and second ports, to operate as a distributed media access control (MAC) bridge in a Layer 2 data network;
1[b]	configuring a link aggregation (LAG) group of parallel physical links between two endpoints in said Layer 2 data network joined together into a single logical link, said LAG group having a plurality of LAG ports and a plurality of conjoined member line cards;
1[c]	providing for each of said member line cards a respective forwarding database (FDB) to hold records associating MAC addresses with ports of said plurality of ports of said network node;
1[d]	receiving a data packet on an ingress port of said network node from a MAC source address, said data packet specifying a MAC destination address on said Layer 2 data network;
1[e]	conveying, by transmitting said data packet to said MAC destination address via said first port, said received data packet in said network node to at least said first line card for transmission to said MAC destination address;
1[f]	if said MAC destination address does not appear in said FDB, flooding said data packet via one and only one LAG port of said plurality of LAG ports;
1[g]	checking said MAC source address of the data packet against records in said FDB of said first line card; and
1[h]	if said FDB of said first line card does not contain a record of an association of said MAC source address with said ingress port, creating a new record of said association, adding said new record to the FDB of said first line card, and sending a message of the association to each member line card of said plurality of member line cards.
2	The method according to claim 1, wherein sending the message comprises sending messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
3	The method according to claim 1, and comprising receiving the message at the second line card, and responsively to the message, adding the record of the association to the FDB of the second line card if the record does not already exist in the FDB of the second line card.
4	The method according to claim 3, and comprising marking the records in the respective FDB of each line card to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
5[pre]	The method according to claim 4, and comprising

No.	Limitation
5[a]	associating a respective aging time with each of the records;
5[b]	refreshing the records in the FDB responsively to further packets transmitted by the line cards; and
5[c]	removing the records from the respective FDB if the records are not refreshed within the respective aging time.
6	The method according to claim 1, wherein sending the message comprises transmitting a synchronization packet from the first line card via a switching core of the network node to at least the second line card
7	The method according to claim 6, wherein sending the synchronization packet comprises, if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, changing the record in the FDB of the first line card and sending a synchronization update packet to at least the second line card to indicate that the record has been changed.
8	The method according to claim 1, wherein the network node is configured to operate as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances
9	The method according to claim 8, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein sending the message comprises identifying the VPLS instance in the message so as to inform all the line cards that serve the VPLS instance.
10[pre]	The method according to claim 1, and comprising:
10[a]	conveying a further data packet, received from a further MAC source address, to the second line card for transmission over the network;
10[b]	checking the further MAC source address against the records in the FDB of the second line card; and
10[c]	responsively to the further data packet, adding a further record with respect to the MAC source address to the FDB of the second line card and sending a further message to inform at least the first line card of the further record.
11[pre]	A node for network communication, comprising:
11[a]	a switching core;
11[b]	a plurality of ports;
11[c]	a plurality of member line cards conjoined in a link aggregation (LAG) group of parallel physical links between two endpoints in a Layer 2 data network joined together into a single logical link, having a plurality of LAG ports to forward packets through said switching core so that the node operates as a virtual media access control (MAC) bridge in said Layer 2 data network, said plurality of

No.	Limitation
	member line cards including at least first and second line cards, each line card having respective ports and having a respective forwarding database (FDB) to hold records associating MAC addresses with said respective ports of said line cards,
11[d]	wherein said line cards are arranged so that upon receiving a data packet on an ingress line card from a MAC source address, said data packet specifying a MAC destination address, said ingress line card conveys said data packet via said switching core to at least said first line card for transmission to said MAC destination address, whereupon said first line card checks said MAC source address of said data packet against records in said FDB of said first line card, and if said FDB database of said first line card does not contain a record of an association of said MAC source address with said ingress port, adds said record to the FDB of said first line card and sends a message to at least said second line card informing said second line card of said association, and arranged, when said MAC destination address does not appear in said FDB, to flood said data packet via one and only one of said LAG ports.
12	The node according to claim 11, wherein at least the first line card is adapted to send messages periodically at predefined times to inform at least the second line card of new associations between the MAC addresses and the respective ports.
13	The node according to claim 11, wherein responsively to the message, the second line card adds the record of the association to the MAC database of the second line card if the record does not already exist in the FDB of the second line card.
14	The node according to claim 13, wherein the records in the respective FDB of each line card are marked to distinguish a first type of the records, which are added in response to data packets transmitted via a port of the line card, from a second type of the records, which are added in response to messages received from another of the line cards.
15	The node according to claim 14, wherein a respective aging time is associated with each of the records, and wherein the line cards are operative to refresh the records in the FDB responsively to further packets transmitted by the line cards, and to remove the records from the respective FDB if the records are not refreshed within the respective aging time.
16	The node according to claim 11, wherein the message comprises a synchronization packet, which is transmitted from the first line card via the switching core to at least the second line card.
17	The node according to claim 16, wherein the line cards are operative so that if the record in the FDB associates the MAC source address with a port different from the one of the ports on which the data packet was received, the record in the FDB of the first line card is changed, and the synchronization packet comprises a synchronization update packet, which indicates to at least the second line card to indicate that the record has been changed.
18	The node according to claim 11, wherein at least some of the line cards are configured so that the node operates as multiple virtual MAC bridges in a Layer 2 virtual private network (VPN), wherein each virtual MAC bridge is configured to

No.	Limitation
	serve a respective VPN instance, and wherein the records associating the MAC addresses with the respective ports are maintained independently for each of the VPN instances.
19	The node according to claim 18, wherein the VPN instance is a VPLS instance among multiple VPLS instances served by the network node, and wherein the VPLS instance is identified in the message so as to inform all the line cards that serve the VPLS instance of the association.
20	The node according to claim 11, wherein the line cards are adapted to forward a further data packet, received from a further MAC source address, to the second line card for transmission over the network, whereupon the second line card checks the further MAC source address against the records in the FDB of the second line card, and responsively to the further data packet, adds a further record with respect to the MAC source address to the FDB of the second line card and sends a further message to inform at least the first line card of the further record.

EXHIBIT 3E

IN THE UNITED STATES DISTRICT COURT

FOR THE WESTERN DISTRICT OF TEXAS

WACO DIVISION

CORRIGENT CORPORATION CIVIL ACTION NO.
6:22-cv-00396-ADA
VS.

CISCO SYSTEMS, INC.

* *

CONFIDENTIAL TRANSCRIPT - ATTORNEYS' EYES ONLY

* *

The remote videotaped deposition of RAMESH PONNAPALLI, taken in connection with the captioned cause, pursuant to the following stipulations before RITA A. DEROUEN, Certified Court Reporter, Registered Professional Reporter, on October 11, 2023, beginning at 11:05 a.m.

A. Yeah. '369, at the fundamental plane, is about checking the idleness of the link. There are other claims, but they're all dependent on this particular claim that you pick an idle link among the LAG members. So it's not just the LAG members; even on regular ports, we never check for link idleness.
Q. So your position is that the claims of the '369 patent require a check for an idle line?
A. Yes.
Q. Okay. Now, what is an idle line?
A. So idle line, in this case, is something where there -- there's no traffic or it has the least amount of traffic.
Q. So an idle line can be a line that has a relatively low amount of traffic?
A. Yes. In this context, that's what I infer.
Q. Okay. Are there any more reasons you're prepared to testify on today that the Nexus 9000 does not infringe the '369 patent?
A. If you have a specific question on any of the claims, I can read out an answer on that.
Q. You said you can read out an answer on that?

EXHIBIT 4A

IEEE Standards for Local and Metropolitan Area Networks:

Virtual Bridged Local Area Networks

Sponsor

LAN MAN Standards Committee
of the
IEEE Computer Society

Approved 8 December 1998

IEEE-SA Standards Board

Abstract: This standard defines an architecture for Virtual Bridged LANs, the services provided in Virtual Bridged LANs, and the protocols and algorithms involved in the provision of those services.
Keywords: local area networks, MAC Bridge management, media access control bridges, virtual LANs

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Print: ISBN 0-7381-1537-1 SH94709
PDF: ISBN 0-7381-1538-X SS94709

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

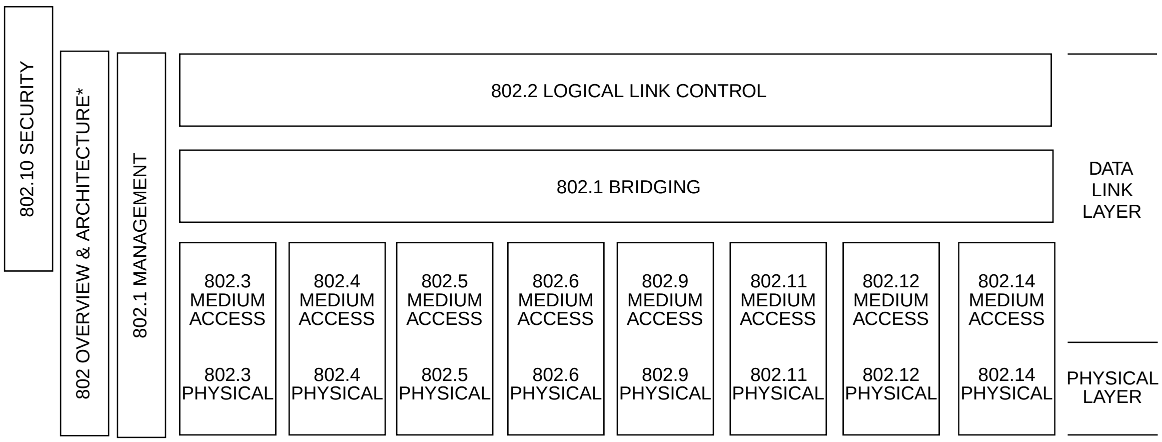
Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Introduction to IEEE Std 802.1Q-1998

(This introduction is not part of IEEE Std 802.1Q-1998, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks.)

This standard is part of a family of standards for local and metropolitan area networks. The relationship between the standard and other members of the family is shown below. (The numbers in the figure refer to IEEE standard numbers.)



* Formerly IEEE Std 802.1A.

This family of standards deals with the Physical and Data Link layers as defined by the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) Basic Reference Model (ISO/IEC 7498-1: 1994). The access standards define seven types of medium access technologies and associated physical media, each appropriate for particular applications or system objectives. Other types are under investigation.

The standards defining the technologies noted above are as follows:

- IEEE Std 802 — *Overview and Architecture.* This standard provides an overview to the family of IEEE 802 Standards.
- ANSI/IEEE Std 802.1B and 802.1k [ISO/IEC 15802-2] — *LAN/MAN Management.* Defines an OSI management-compatible architecture, and services and protocol elements for use in a LAN/MAN environment for performing remote management.
- ANSI/IEEE Std 802.1D [ISO/IEC 15802-3] — *Media Access Control (MAC) Bridges.* Specifies an architecture and protocol for the interconnection of IEEE 802 LANs below the MAC service boundary.
- ANSI/IEEE Std 802.1E [ISO/IEC 15802-4] — *System Load Protocol.* Specifies a set of services and protocol for those aspects of management concerned with the loading of systems on IEEE 802 LANs.
- ANSI/IEEE Std 802.1F — *Common Definitions and Procedures for IEEE 802 Management Information*
- ANSI/IEEE Std 802.1G [ISO/IEC 15802-5] — *Remote Media Access Control (MAC) bridging.* Specifies extensions for the interconnection, using non-LAN communication technologies, of geographically separated IEEE 802 LANs below the level of the logical link control protocol.
- ANSI/IEEE Std 802.2 [ISO/IEC 8802-2] — *Logical link control*
- ANSI/IEEE Std 802.3 [ISO/IEC 8802-3] — *CSMA/CD access method and physical layer specifications*

- ANSI/IEEE Std 802.4 [ISO/IEC 8802-4] — *Token passing bus access method and physical layer specifications*
- ANSI/IEEE Std 802.5 [ISO/IEC 8802-5] — *Token ring access method and physical layer specifications*
- ANSI/IEEE Std 802.6 [ISO/IEC 8802-6] — *Distributed Queue Dual Bus (DQDB) access method and physical layer specifications*
- ANSI/IEEE Std 802.9 [ISO/IEC 8802-9] — *Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers*
- ANSI/IEEE Std 802.10 — *Interoperable LAN/MAN Security*
- ANSI/IEEE Std 802.11 [ISO/IEC DIS 8802-11] — *Wireless LAN Medium Access Control (MAC) and physical layer specifications*
- ANSI/IEEE Std 802.12 [ISO/IEC 8802-12] — *Demand-priority access method, physical layer and repeater specifications*

In addition to the family of standards, the following is a recommended practice for a common Physical Layer technology:

- IEEE Std 802.7 — *IEEE Recommended Practice for Broadband Local Area Networks*

The following additional working group has authorized standards projects under development:

- IEEE 802.14 — *Standard Protocol for Cable-TV Based Broadband Communication Network*

Conformance test methodology

An additional standards series, identified by the number 1802, has been established to identify the conformance test methodology documents for the 802 family of standards. Thus, the conformance test documents for 802.3 are numbered 1802.3.

IEEE Std 802.1Q-1998

The MAC Bridge standardization activities that resulted in the development of ISO/IEC 10038: 1993 introduced the concept of Filtering Services in Bridged LANs, and mechanisms whereby filtering information in such LANs may be acquired and held in a Filtering Database.

ISO/IEC 15802-3, a revision of ISO/IEC 10038, extends this concept of Filtering Services in order to define additional capabilities in Bridged LANs aimed at

a) The provision of expedited traffic capabilities, to support the transmission of time-critical information in a LAN environment;

b) The use of signalled user priority information as the basis for identifying expedited classes of traffic;

c) The provision of filtering services that support the dynamic definition and establishment of Groups in a LAN environment, and the filtering of frames by Bridges such that frames addressed to a particular Group are forwarded only on those LAN segments that are required in order to reach members of that Group;

d) The provision of a Generic Attribute Registration Protocol (GARP) that is used to support the mechanism for providing Group filtering capability, and is also made available for use in other attribute registration applications.

This standard makes use of the concepts and mechanisms of LAN Bridging that were introduced by ISO/IEC 15802-3, and defines additional mechanisms that allow the implementation of Virtual Bridged LANs. The following are described:

e) Virtual LAN Services in Bridged LANs;

f) The operation of the Forwarding Process that is required in order to support Virtual Bridged LANs;

g) The structure of the Filtering Database that is required in order to support Virtual Bridged LANs;

h) The nature of the protocols and procedures that are required in order to provide Virtual LAN services, including the definition of the frame formats used to represent VLAN identification information, and the procedures used in order to insert and remove VLAN identifiers and the headers in which they are carried;

i) The ability to support end-to-end signalling of user priority information regardless of the intrinsic ability of the underlying MAC protocols to signal user priority information;

j) The GARP VLAN Registration Protocol (GVRP) that allows distribution and registration of VLAN membership information (the protocol described makes use of the GARP protocol defined in ISO/IEC 15802-3);

k) The management services and operations that are required in order to configure and administer Virtual Bridged LANs.

This standard contains state-of-the-art material. The area covered by this standard is undergoing evolution. Revisions are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Information on the current revision state of this and other IEEE 802 standards may be obtained from

Secretary, IEEE-SA Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

IEEE 802 committee working documents are available from

IEEE Document Distribution Service
AlphaGraphics #35 Attn: P. Thrush
10201 N. 35th Avenue
Phoenix, AZ 85051
USA

Participants

The following is a list of participants in the Interworking activities of the IEEE 802.1 Working Group. Voting members at the time of publication are marked with an asterisk (*).

William P. Lidinsky, *Chair**
Mick Seaman, *Chair, Interworking Task Group**
Editing Team:
Tony Jeffree*, *Coordinating Editor*
Anil Rijsinghani*, **Richard Hausman***, **Michele Wright***, **Paul Langille***, **P. J. Singh***

Steve Adams*
Stephen Ades
Ken Alonge
Floyd Backes*
John Bartlett*
Les Bell*
Avner Ben-Dor
Michael Berger*
James S. Binder*
David Brady
Martin Brewer
Bill Bunch*
Bob Cardinal
Paul Carroll*
Jeffrey Catlin*
Dennis Cave
Alan Chambers*
Steve Chan
David W. Chang*
Ken Chapman
Hon Wah Chin*
Chi Chong
Chris Christ*
Paul Congdon*
Glenn Connery*
David Cullerot*
Ted Davies*
Andy Davis
David Delaney*
Prakash Desai
Jeffrey Dietz*
Kurt Dobbins
Peter Ecclesine*
J. J. Ekstrom*
Norman W. Finn*
Yishai Fraenkel
Paul Frantz
Lars Henrik Frederiksen*
Anoop Ghanwani*
John Grinham
Steve Haddock
Sharam Hakimi*
John Hart*
Scott Harvell
Wayne Hathaway
Vic Hayes
David Head*
Gaby Hecht
Deepak Hegde*
Ariel Hendel
John Hickey
David Hollender
Steve Horowitz*
Michelle Hsiung
Rita Hunt
David Husak
Altaf Hussain*
Vipin K. Jain*
Neil Jarvis
Allen Kasey
Toyoyuki Kato*
Hal Keen*
Kevin Ketchum*
Keith Klamm*
Bruce Kling*
Walter Knitl
Dan Krent*
Paul Kummer
Paul Lachapelle*
Bill Lane
Johann Lindmeyr*
Gary Littleton
Robert D. Love
Andy Luque
Peter Martini
Keith McCloghrie
Martin McNealis
Milan Merhar*
John Messenger*
Colin Mick
Amol Mitra
Yaron Nachman*
Krishna Narayanaswamy*
Paul Nikolich
Lawrence Ng*
Henry Ngai*
Eugene O'Neil
Satoshi Obara*
Toshio Ooka*
Jörg Ottensmeyer*
Luc Pariseau*
Yonadav Perry
John Pickens*
Gideon Prat
Kirk Preiss
Steve Ramberg*
Shlomo Reches*
Dick Reohr
James Richmond*
Doug Ruby
Ray Samora
Ayman Sayed*
Rich Seifert
Lee Sendelbach*
Himanshu Shah*
Phil Simmons*
K. Karl Shimada
Fred Shu
Rosemary V. Slager*
Alexander Smith*
Andrew Smith*
Larry Stefani*
Stuart Soloway*
Sundar Subramaniam*
Richard Sweatt
Robin Tasker*
Fouad Tobagi
Naoki Tsukutari
Dhadesugoor Vaman
Steve Van Seters*
Dono van-Mierop*
John Wakerly*
Peter Wang*
Philip Wang
Y. C. Wang*
Trevor Warwick*
Bob Watson
Alan Weissberger
Glenn Wenig
Keith Willette*
Michael Witkowski*
Edward Wong*
Michael D. Wright*
Allen Yu*
Wayne Zakowski*

The following persons were on the balloting committee of IEEE Std 802.1Q:

Corey Anderson
Kit Athul
Thomas W. Bailey
Peter K. Campbell
James T. Carlo
David E. Carlson
Alan M. Chambers
R. Allan Cobb
Robert S. Crowder
Thomas J. Dineen
Peter Ecclesine
Philip H. Enslow
Changxin Fan
John W. Fendrich
Michael A. Fischer
Harvey A. Freeman
Gautam Garai
Harry Gold
Julio Gonzalez-Sanz
Stephen R. Haddock
Allen W. Hathaway
Donald N. Heirman
Raj Jain
Neil A. Jarvis
Anthony A. Jeffree
Robert W. Klessig
Stephen Barton Kruger
William G. Lane
David J. Law
Lanse M. Leach
Randolph S. Little
Peter Martini
Milan Merhar
John L. Messenger
Bennett Meyer
David S. Millman
John E. Montague
Wayne D. Moyers
Shimon Muller
Paul Nikolich
Charles Oestereicher
Roger Pandanda
John R. Pickens
Vikram Punj
Edouard Y. Rocher
James W. Romlein
Floyd E. Ross
Michael Salzman
Norman Schneidewind
Mick Seaman
Rich Seifert
Leo Sintonen
Michael A. Smith
Patricia Thaler
Geoffrey O. Thompson
Mark-Rene Uchida
Oren Yuen

When the IEEE-SA Standards Board approved this standard on 8 December 1998, it had the following membership:

Richard J. Holleman, *Chair* **Donald N. Heirman,** *Vice Chair*
Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon
James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. “Al” Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry
L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Kristin M. Dittmann
IEEE Standards Project Editor

Contents

Figures

Tables

IEEE Standards for Local and Metropolitan Area Networks:

Virtual Bridged Local Area Networks

1. Overview

IEEE 802 Local Area Networks (LANs) of all types can be connected together with Media Access Control (MAC) Bridges, as specified in ISO/IEC 15802-3[1]. This standard defines the operation of Virtual LAN (VLAN; see 3.16) Bridges that permit the definition, operation, and administration of VLAN topologies within a Bridged LAN (see 3.20) infrastructure.

1.1 Scope

For the purpose of compatible interconnection of information technology equipment using the IEEE 802 MAC Service supported by interconnected IEEE 802 standard LANs using different or identical MAC methods, this standard specifies a general method for the operation of MAC Bridges that support the construction of VLANs (see 3.16). To this end it

a) Positions the function of VLANs within an architectural description of the MAC Sublayer;
b) Defines enhancements to the Support of the MAC Service, as described and defined in ISO/IEC 15802-3, for the purposes of VLAN Bridging;
c) Specifies an Enhanced Internal Sublayer Service provided to the Media Access Independent functions that provide frame relay (3.4) in the VLAN Bridge;
d) Specifies the operation of the functions that provide frame relay in the VLAN Bridge;
e) Defines the structure, encoding, and interpretation of the VLAN control information carried in tagged frames (3.12) in a VLAN;
f) Specifies the rules that govern the insertion and removal of VLAN control information in frames;
g) Specifies the rules that govern the ability to carry data in Canonical format and Non-canonical format using different LAN MAC methods;

NOTE—The meanings of the terms *Canonical format* and *Non-canonical format* are discussed in Annex F.

h) Establishes the requirements for, and specifies the means of, automatic configuration of VLAN topology information;
i) Defines the management functionality that may be provided in a VLAN Bridge in order to facilitate administrative control over VLAN operation;
j) Specifies the requirements to be satisfied by equipment claiming conformance to this standard.

1.2 VLAN aims and benefits

VLANs aim to offer the following benefits:

a) VLANs are supported over all IEEE 802 LAN MAC protocols, and over shared media LANs as well as point-to-point LANs.

[1]Information about references can be found in Clause 2.

b) VLANs facilitate easy administration of logical groups of stations that can communicate as if they were on the same LAN. They also facilitate easier administration of moves, adds, and changes in members of these groups.
c) Traffic between VLANs is restricted. Bridges forward unicast, multicast, and broadcast traffic only on LAN segments that serve the VLAN to which the traffic belongs.
d) As far as possible, VLANs maintain compatibility with existing bridges and end stations.
e) If all Bridge Ports are configured to transmit and receive untagged frames (3.14), bridges will work in plug-and-play ISO/IEC 15802-3 mode. End stations will be able to communicate throughout the Bridged LAN.

NOTE—Whether a VLAN Bridge will operate in ISO/IEC 15802-3 mode depends upon the configuration of the various Port parameters (8.4) and the Filtering Database (8.11). A VLAN Bridge in its default configuration is transparent to untagged frames (3.14) but is not transparent to tagged frames (3.12), so the operation of such Bridges in the presence of tagged traffic differs from that of an ISO/IEC 15802-3 Bridge. If the configuration settings of VLAN Bridges are changed from the default values defined in this standard, then transparency with respect to untagged frames may also be affected.

1.3 Relationship with ISO/IEC 15802-3

This standard makes use of specific aspects of the MAC Bridge specification contained in ISO/IEC 15802-3; those aspects therefore become provisions of this standard. Table 1-1 shows how the relevant clauses of ISO/IEC 15802-3 are incorporated into this standard.

Table 1-1—Relationship between this standard and ISO/IEC 15802-3

ISO/IEC 15802-3 clause	Use in this standard
5. Conformance	Provision of this standard, as extended by Clause 5
6. Support of the MAC Service	Provision of this standard, as extended by Clause 7
7. Principles of operation	Replaced by Clause 8
8. The spanning tree algorithm and protocol	Provision of this standard
9. Encoding of bridge protocol data units	Provision of this standard
10. GARP Multicast Registration Protocol (GMRP)	Provision of this standard, as modified by Clause 11
11. Example “C” code implementation of GMRP	Provision of this standard, as modified by Clause 11
12. Generic Attribute Registration Protocol (GARP)	Provision of this standard
13. Example “C” code implementation of GARP	Provision of this standard
14. Bridge management	Replaced by Clause 12
15. Management protocol	Not applicable
16. Bridge performance	Provision of this standard
Annex A (normative) PICS proforma	Replaced by Annex A
Annex B (informative) Calculating Spanning Tree parameters	Provision of this standard
Annex C (normative) Source-routing transparent bridge operation	Provision of this standard
Annex D (normative) PICS proforma for source routing transparent bridge operation	Provision of this standard
Annex E (normative) Allocation of Object Identifier values	Not applicable
Annex F (informative) Target topology, migration, and interoperability	Provision of this standard
Annex G (informative) Preserving the integrity of FCS fields in MAC Bridges	Provision of this standard
Annex H (informative) Design considerations for Traffic Class Expediting and Dynamic Multicast Filtering	Provision of this standard

2. References

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ANSI X3.159-1989, American National Standards for Information Systems—Programming Language—C.[2]

IEEE Std 802-1990, IEEE Standards for Local and Metropolitan Area Networks: Overview and Architecture.[3]

IEEE Std 802.1F-1993, IEEE Standards for Local and Metropolitan Area Networks: Common Definitions and Procedures for IEEE 802 Management Information.

IEEE Std 802.3, 1998 Edition, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

IEEE Std 802.3ac-1998, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Supplement to Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications: Frame Extensions for Virtual Bridged Local Area Network (VLAN) tagging on 802.3 Networks.

IEEE Std 802.9a-1995, IEEE Standards for Local and Metropolitan Area Networks: Supplement to Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers: Specification of ISLAN 16-T.

IETF RFC 1042, Postel & Reynolds, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks, February 1988.[4]

IETF RFC 1390, D. Katz, Transmission of IP and ARP over FDDI Networks, January 1993.

ISO 6937-2: 1994, Information technology—Coded graphic character set for text communication—Latin alphabet.[5]

ISO/IEC 7498-1: 1994, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 1: The Basic Model.

ISO/IEC 7498-4: 1989, Information processing systems—Open Systems Interconnection—Basic Reference Model—Part 4: Management framework.

---

[2]ANSI publications are available from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

[3]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

[4]Internet RFCs are retrievable by FTP at ds.internic.net/rfc/rfcnnnn.txt (where nnnn is a standard's publication number such as 1042), or call InterNIC at 1-800-444-4345 for information about receiving copies through the mail.

[5]ISO and ISO/IEC documents are available from the ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse; and from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

ISO/IEC 8802-2: 1998 [ANSI/IEEE Std 802.2, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical link control.[6]

ISO/IEC 8802-4: 1990 [ANSI/IEEE Std 802.4-1990], Information processing systems—Local area networks—Part 4: Token-passing bus access method and physical layer specifications.

ISO/IEC 8802-5: 1998 [ANSI/IEEE Std 802.5, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 5: Token ring access method and physical layer specifications.

ISO/IEC 8802-6: 1994 [ANSI/IEEE Std 802.6, 1994 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 6: Distributed Queue Dual Bus (DQDB) access method and physical layer specifications.

ISO/IEC 8802-9: 1996 [ANSI/IEEE Std 802.9, 1996 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 9: Integrated Services (IS) LAN Interface at the Medium Access Control (MAC) and Physical (PHY) Layers.

ISO/IEC DIS 8802-11, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

ISO/IEC 8802-12: 1998 [ANSI/IEEE Std 802.12, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 12: Demand-Priority access method, physical layer and repeater specifications.

ISO/IEC 8824: 1990, Information technology—Open Systems Interconnection—Specification of Abstract Syntax Notation One (ASN.1) (Provisionally retained edition).

ISO/IEC 8825: 1990, Information technology—Open Systems Interconnection—Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1) (Provisionally retained edition).

ISO 9314-2: 1989, Information processing systems—Fibre Distributed Data Interface—Part 2: FDDI Token Ring Media Access Control (MAC).

ISO/IEC 9595: 1998, Information technology—Open Systems Interconnection—Common management information service.

ISO/IEC 9596: 1998, Information technology—Open Systems Interconnection—Common management information protocol—Part 1: Specification.

ISO/IEC 10038: 1993 [ANSI/IEEE Std 802.1D-1993], Information technology—Telecommunications and information exchange between systems—Local area networks—Media Access Control (MAC) bridges.

ISO/IEC 11802-5: 1997 [ANSI/IEEE Std 802.1H-1997], Information technology—Telecommunications and information exchange between systems—Local and Metropolitan Area Networks—Technical Reports and Guidelines—Part 5: Media Access Control Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks.

[6]ISO [IEEE] and ISO/IEC [IEEE] documents are available from ISO Central Secretariat, 1 rue de Varembé, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse; and from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

ISO/IEC 15802-1: 1995, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 1: Medium Access Control (MAC) service definition.

ISO/IEC 15802-2: 1995 [ANSI/IEEE Std 802.1B, 1995], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 2: LAN/MAN Management.

ISO/IEC 15802-3: 1998 [ANSI/IEEE Std 802.1D, 1998 Edition], Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Common specifications—Part 3: Media Access Control (MAC) Bridges.

3. Definitions

The following terms are specific to this standard:

3.1 Ethernet Type-encoding

The use of the Type interpretation of an IEEE 802.3 Length/Type field value in a frame as a protocol identifier associated with the MAC Service user data carried in the frame.[7]

NOTES

1—The term *frame* is defined in 3.3.

2—Ethernet Type-encoding can be used with MAC Service user data carried on non-IEEE 802.3 MACs by means of the SNAP-based encapsulation techniques specified in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.

3.2 Logical Link Control (LLC) encoding

The use of LLC addressing information in a frame as a protocol identifier associated with the MAC Service user data carried in the frame.

3.3 Frame

A unit of data transmission on an IEEE 802 LAN MAC that conveys a protocol data unit (PDU) between MAC Service users. There are three types of frame; *untagged*, *VLAN-tagged*, and *priority-tagged*.

NOTE—The term *IEEE 802 LAN* is defined in ISO/IEC 15802-3. *Untagged frame* is defined in 3.14, *VLAN-tagged frame* is defined in 3.18, and *priority-tagged frame* is defined in 3.8.

3.4 Frame relay

The function of the Forwarding Process that forwards frames between the Ports of a Bridge.

NOTE—The operation of the Forwarding Process is defined in 8.7.

3.5 Independent Virtual Local Area Network (VLAN) Learning (IVL)

Configuration and operation of the Learning Process and the Filtering Database such that, for a given set of VLANs, if a given individual MAC Address is learned in one VLAN, that learned information is not used in forwarding decisions taken for that address relative to any other VLAN in the given set.

NOTE—In a Bridge that supports only IVL operation, the “given set of VLANs” is the set of all VLANs.

3.6 Independent Virtual Local Area Network (VLAN) Learning (IVL) Bridge

A type of Bridge that supports only Independent VLAN Learning.

[7]The use of Ethernet Type values as a means of protocol identification was defined in the specification of Ethernet V2.0 (The Ethernet, AA-K759B-TK, Digital Equipment, Intel, and Xerox Corps., Nov. 1982).

3.7 Legacy region

A set of LAN segments interconnected such that there is physical connectivity between any pair of segments using only ISO/IEC 15802-3-conformant, VLAN-unaware MAC Bridges.

NOTE—In other words, if, in a Bridged LAN containing both ISO/IEC 15802-3 and IEEE 802.1Q Bridges, all the IEEE 802.1Q Bridges were to be removed, the result would be a set of one or more Bridged LANs, each with its own distinct Spanning Tree. Each of those Bridged LANs is a legacy region. The term VLAN-unaware is defined in 3.19.

3.8 Priority-tagged frame

A tagged frame whose tag header carries priority information, but carries no VLAN identification information.

NOTE—The term *tagged frame* is defined in 3.12; *tag header* is defined in 3.13. See also *VLAN-tagged frame* (3.18) and *untagged frame* (3.14).

3.9 Shared Virtual Local Area Network (VLAN) Learning (SVL)

Configuration and operation of the Learning Process and the Filtering Database such that, for a given set of VLANs, if an individual MAC Address is learned in one VLAN, that learned information is used in forwarding decisions taken for that address relative to all other VLANs in the given set.

NOTE—In a Bridge that supports only SVL operation, the "given set of VLANs" is the set of all VLANs.

3.10 Shared Virtual Local Area Network (VLAN) Learning (SVL) Bridge

A type of Bridge that supports only Shared VLAN Learning.

3.11 Shared Virtual Local Area Network (VLAN) Learning (SVL)/Independent Virtual Local Area Network (VLAN) Learning (IVL) Bridge

An SVL/IVL Bridge is a type of Bridge that simultaneously supports both Shared VLAN Learning and Independent VLAN Learning.

3.12 Tagged frame

A *tagged frame* is a frame that contains a tag header immediately following the Source MAC Address field of the frame or, if the frame contained a Routing Information field, immediately following the Routing Information field. There are two types of tagged frames: VLAN-tagged frames and priority-tagged frames.

NOTE—The term *tag header* is defined in 3.13, *priority-tagged frame* is defined in 3.8, and *VLAN-tagged frame* is defined in 3.18. See also *untagged frame* (3.14).

3.13 Tag header

A tag header allows user priority information, and optionally, VLAN identification information, to be associated with a frame.

NOTE—The structure of the tag header is defined in 9.3.

3.14 Untagged frame

An *untagged frame* is a frame that does not contain a tag header immediately following the Source MAC Address field of the frame or, if the frame contained a Routing Information field, immediately following the Routing Information field.

NOTE—The term *tag header* is defined in 3.13. See also *tagged frame* (3.12).

3.15 Virtual Bridged Local Area Network (LAN)

A Bridged LAN in which the existence of one or more VLAN-aware Bridges allows the definition, creation, and maintenance of VLANs.

NOTE—The term *VLAN-aware* is defined in 3.17.

3.16 Virtual Local Area Network (VLAN)

A subset of the active topology of a Bridged Local Area Network. Associated with each VLAN is a VLAN Identifier (VID).

3.17 VLAN-aware

A property of Bridges or end stations that recognize and support VLAN-tagged frames.

3.18 VLAN-tagged frame

A tagged frame whose tag header carries both VLAN identification and priority information.

3.19 VLAN-unaware

A property of Bridges or end stations that do not recognize VLAN-tagged frames.

3.20 Terms used in ISO/IEC 15802-3

The following terms used in this standard are used in ISO/IEC 15802-3:

Active topology
Bridge Port
Bridged Local Area Network (also Bridged LAN)
GARP Participant
GARP Application
GIP Context
Group
IEEE 802 Local Area Network (also IEEE 802 LAN, or LAN)
Port

4. Abbreviations

The following abbreviations are used in this standard:

BPDU	Bridge Protocol Data Unit (ISO/IEC 15802-3)
CFI	Canonical Format Indicator (Annex F)
E-ISS	Enhanced Internal Sublayer Service (7.1)
FCS	Frame Check Sequence (7.1)
FID	Filtering Identifier (8.11.3, 8.11.7)
GARP	Generic Attribute Registration Protocol (ISO/IEC 15802-3)
GID	GARP Information Declaration (ISO/IEC 15802-3)
GIP	GARP Information Propagation (ISO/IEC 15802-3)
GMRP	GARP Multicast Registration Protocol (Clause 10, ISO/IEC 15802-3)
GVRP	GARP VLAN Registration Protocol (Clause 11)
ISS	Internal Sublayer Service (Clause 7, ISO/IEC 15802-3)
IVL	Independent VLAN Learning (3.5)
LAN	Local Area Network (IEEE Std. 802)
LS	Least-significant
LLC	Logical Link Control (ISO/IEC 8802-2)
MAC	Medium Access Control (IEEE Std. 802)
MIB	Management Information Base (ISO/IEC 7498-4)
MS	Most-significant
MSDU	MAC Service Data Unit (ISO/IEC 15802-1)
NCFI	Non-Canonical Format Indicator (Annex F)
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement (Annex A)
PVID	Port VID (8.4.4)
RIF	Routing Information Field (ISO/IEC 8802-5)
STPID	SNAP-encoded Tag Protocol Identifier (9.3)
SVL	Shared VLAN Learning (3.9)
TCI	Tag Control Information (9.3)
TPID	Tag Protocol Identifier (9.3)
VID	VLAN Identifier (7.1, 8.4.4, 9.3)
VLAN	Virtual LAN (3.16)

5. Conformance

5.1 Static conformance requirements

A MAC Bridge for which conformance to this standard is claimed shall

a) Conform to the requirements of ISO/IEC 15802-3, as modified by the provisions of this standard;
b) Relay and filter frames as described in 8.1 and specified in 8.5, 8.6, 8.7, 8.8, and 8.9;
c) On each Port, support at least one of the permissible values for the Acceptable Frame Types parameter, as defined in 8.4.3;
d) Support the following on each Port that supports untagged and priority-tagged frames:
   1) A Port VLAN Identifier (PVID) value (8.4.4);
   2) The ability to configure at least one VLAN whose untagged set includes that Port (8.8 and 8.11.9);
   3) Configuration of the PVID value via management operations (12.10);
   4) Configuration of Static Filtering Entries via management operations (12.7).
e) Support the ability to insert tag headers into, modify tag headers in, and remove tag headers from relayed frames, as described and specified in 7.1 and Clause 9, as required by the value(s) of the Acceptable Frame Types parameter that are supported on each Port, and by the ability of each Port to be configured to transmit VLAN-tagged frames and/or untagged frames. These requirements are summarized in Table 5-1 for frames relayed between any pair of Ports;

Table 5-1—Support requirements for insertion, removal, and modification of tag headers

		Reception Port receives as (and does not discard):		
		VLAN-tagged	**Priority-tagged**	**Untagged**
Transmission Port transmits as:	**Untagged**	Shall support removal of tag headers.	Shall support removal of tag headers.	N/A.
	VLAN-tagged	Shall support conversion of the tagged frame format if the format required for the destination MAC differs from the received format.	Shall support the insertion of a non-null Virtual LAN Identifier (VID) in tag headers, plus conversion of the tagged frame format if the format required for the destination MAC differs from the received format.	Shall support the insertion of tag headers of a format appropriate to the destination MAC, carrying a non-null VID.

f) Support the ability to perform automatic configuration and management of VLAN topology information by means of Generic Attribute Registration Protocol (GARP) VLAN Registration Protocol (GVRP) (Clause 11) on all Ports;
g) Support the ability for the Filtering Database to contain static and dynamic configuration information for at least one VLAN, by means of Static and Dynamic VLAN Registration Entries (8.11);
h) Support at least one Filtering Identifier (FID) (6.4, 8.11.3, 8.11.7, and 8.11.8);
i) Support the ability to allocate at least one VID to each FID that is supported (6.4, 8.11.3, 8.11.7, and 8.11.8).

NOTE—Under some circumstances, the ability for VLAN Bridges to successfully interoperate depends upon the number of FIDs supported, and the number of VIDs that can be allocated to each FID. These circumstances are discussed in Annex B, along with the implications with respect to interoperability.

5.2 Options

A MAC Bridge for which conformance to this standard is claimed may

a) Support operation in Extended Filtering Mode (ISO/IEC 15802-3, 6.6.5) and the operation of GARP Multicast Registration Protocol (GMRP) (ISO/IEC 15802-3, Clause 10) as modified by Clause 10;
b) Support the ability for the Filtering Database to contain static and dynamic configuration information for more than one VLAN, by means of Static and Dynamic VLAN Registration Entries (8.11), up to a maximum of 4094 VLANs;

NOTE—The maximum number of VLANs that can be supported is 4094 rather than 4096, as the VID values 0 and FFF are reserved, as indicated in Table 9-2. As conformance to this standard is only with regard to externally visible protocol behavior, this limit on the number of VLANs that can be supported does not imply any such limitation with regard to the internal architecture of a Bridge.

c) On each Port, support both of the permissible values for the Acceptable Frame Types parameter, as defined in 8.4.3. If both values are supported, then the implementation shall support configuration of the parameter value via management;
d) Support the ability to enable and disable Ingress Filtering (8.4.5);
e) Support the ability to configure more than one VLAN whose untagged set includes that Port (8.8 and 8.11.9);
f) Support the management functionality defined in Clause 12;
g) Support more than one FID (6.4, 8.11.3, 8.11.7, and 8.11.8);
h) Support the ability to allocate more than one VID to each FID that is supported (6.4, 8.11.3, 8.11.7, and 8.11.8);
i) Support the ability to configure VLAN Learning Constraints via management (8.11.7 and 12.10.3);
j) Support the ability to configure fixed VID to FID allocations via management (8.11.7.1 and 12.10.3);
k) Support any other optional capabilities defined in ISO/IEC 15802-3, as modified by the provisions of this standard.

5.3 Protocol Implementation Conformance Statement (PICS)

The supplier of an implementation that is claimed to conform to this standard shall complete a copy of the PICS proforma provided in Annex A and shall provide the information necessary to identify both the supplier and the implementation.

5.4 MAC-specific bridging methods

MAC-specific bridging methods may exist. Use of a MAC-specific bridging method and the method specified in this standard on the same LAN shall

a) Not prevent communication between stations in a Bridged LAN.
b) Preserve the MAC Service.
c) Preserve the characteristics of each bridging method within its own domain.
d) Provide for the ability of both bridging techniques to coexist simultaneously on a LAN without adverse interaction.

ISO/IEC 15802-3, Annex C, defines one such MAC-specific bridging method, source routing, and that method is also a provision of this standard. While this standard defines how source-routed frames can be transported in a VLAN environment, it does not attempt to specify VLAN aspects of the source routing bridging method itself.

6. Architectural overview

The architectural framework for VLANs is based on a three-layer model, consisting of the following layers:

a) Configuration;
b) Distribution of configuration information;
c) Relay.

The model is illustrated in Figure 6-1, and further described in the following subclauses.

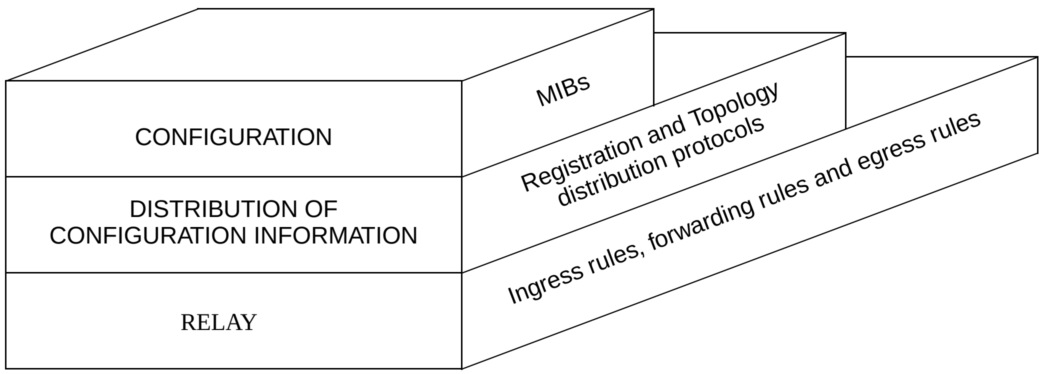


Figure 6-1—VLAN architectural framework

6.1 Configuration

Configuration is concerned with the following issues:

a) The means whereby the VLAN configuration is specified in the first place. This might be achieved via local and/or remote management mechanisms, via server mechanisms, via distribution protocols, or via other means. Determination of the configuration is outside the scope of this standard.
b) Assignment of VLAN configuration parameters.

Clause 12 defines the management operations that are standardized for use in the configuration of VLAN devices.

6.2 Distribution of configuration information

This is the process that allows information to be distributed in order for Bridges to be able to determine to which VLAN a given frame should be classified. Clause 11 defines a general mechanism for the distribution of VLAN membership information to all VLAN-aware devices in a Bridged LAN.

6.3 Relay

This is concerned with the mechanics of

a) Classifying each received frame as belonging to one and only one VLAN. This aspect of relay is determined by a set of MAC Bridge *ingress rules*;

b) Decisions related to where received frames should be forwarded. This aspect of relay is determined by a set of MAC Bridge *forwarding rules*;
c) Mapping frames for transmission through the appropriate outbound Ports, and in appropriate (VLAN-tagged or untagged) format. These aspects of relay are determined by a set of MAC Bridge *egress rules*;
d) The procedures used in order to add, modify, and remove tag headers, when relaying frames between LAN segments, in accordance with the details of the VLAN frame format (defined in Clause 9) used to carry VIDs (otherwise referred to as VLAN tags).

Clause 8 defines ingress, forwarding and egress rules, constituting a generic approach to the provision of VLAN functionality with respect to received VLAN-tagged frames, and a Port-based approach to the VLAN classification of received priority-tagged and untagged frames. Clause 9 defines the format of the tag headers for different MAC methods, and the procedures for adding, modifying, and removing tag headers.

The ingress, forwarding, and egress rules allow Bridges to

e) Classify any received untagged frames or priority-tagged frames that are to be submitted to the Forwarding Process as belonging to a particular VLAN, as defined by the PVID for the receiving Port. The default PVID is specified in Table 9-2;

NOTE 1—This classification of untagged and priority-tagged frames is part of the functionality of the MAC relay entity (Figure 8-3, Figure 8-4), and is therefore only of significance for received frames that are potentially to be forwarded through other Ports of the Bridge (see 6.7).

f) Classify any received VLAN-tagged frames that are to be submitted to the Forwarding Process as belonging to the VLAN identified by the VID carried in the tag header;
g) Make use of the VLAN classification thus associated with the received frame in order to take appropriate forwarding/filtering decisions;
h) Transmit frames in VLAN-tagged or untagged format, as defined for a given Port/VLAN pairing.

NOTE 2—This standard defines a default Port-based classification for VLANs implemented using the procedures and VLAN frame format specified herein. End stations that transmit VLAN-tagged frames, and in the future, Bridges capable of other classification methods, may actually do much of the VLAN classification of frames. More sophisticated tagging will be the rule for these devices, and bridges conformant to this standard will work with them. In this scenario, most or all LAN segments are likely to carry VLAN-tagged frames belonging to various VLANs, but each such LAN segment has its own “local default” VLAN. This “local default” defines the VLAN to which untagged or priority-tagged frames are presumed to belong when received on the Ports of IEEE 802.1Q conformant bridges attached to that LAN segment.

6.4 Filtering Database architecture

The Filtering Database architecture defined in this standard recognizes that

a) For some configurations, it is necessary to allow address information learned in one VLAN to be shared among a number ofVLANs. This is known as *Shared VLAN Learning* (3.9);
b) For some configurations, it is desirable to ensure that address information learned in one VLAN is not shared with other VLANs. This is known as *Independent VLAN Learning* (3.5);
c) For some configurations, it is immaterial as to whether learned information is shared between VLANs or not.

NOTE 1—Annex B discusses the need for Shared and Independent VLAN Learning, and also some of the related interoperability issues.

Shared VLAN Learning is achieved by including learned information from a number of VLANs in the same Filtering Database; Independent VLAN Learning is achieved by including information from each VLAN in distinct Filtering Databases.

NOTE 2—The actual Filtering Database specification specifies a single Filtering Database that, through the inclusion of VLAN identification information in each database entry, can model the existence of one or more distinct Filtering Databases.

Within a given VLAN-Bridged LAN, there may be a combination of configuration requirements, so that individual VLAN Bridges may be called upon to share learned information, or not share it, according to the requirements of particular VLANs or groups of VLANs. The Filtering Database structure that is defined in this standard allows both Shared and Independent VLAN Learning to be implemented within the same VLAN Bridge; i.e., allows learned information to be shared between those VLANs for which Shared VLAN Learning is necessary, while also allowing learned information not to be shared between those VLANs for which Independent VLAN Learning is necessary. The precise requirements for each VLAN with respect to sharing or independence of learned information (if any) are made known to VLAN Bridges by means of a set of *VLAN Learning Constraints* (8.11.7.2), which may be configured into the Bridges by means of management operations. By analyzing the set of learning constraints for the VLANs that are currently active, the Bridge can determine

d) How many independent Filtering Databases are required in order to meet the constraints;
e) For each VLAN, which Filtering Database it will feed any learned information into (and use learned information from).

The manner in which this mapping of VLANs onto Filtering Databases is achieved is defined in 8.11.7; the result is that each VLAN is associated with exactly one Filtering Database.

The most general application of the Filtering Database specification in this standard is a Bridge that can support M independent Filtering Databases, and can map N VLANs onto each Filtering Database. Such a Bridge is known as an SVL/IVL Bridge (3.11).

The conformance requirements in this standard (5.1, 5.2) recognize that VLAN Bridges will be implemented with differing capabilities in order to meet a wide range of application needs, and that the full generality of the SVL/IVL approach is not always either necessary or desirable, as observed in the discussion in Annex B. In a given conformant implementation, there may be restrictions placed upon the number of Filtering Databases that can be supported, and/or the number of VLANs that can be mapped onto each Filtering Database. The full spectrum of conformant Filtering Database implementations is therefore as follows:

f) The SVL/IVL Bridge, as described above. Such Bridges provide support for M Filtering Databases, with the ability to map N VLANs onto each one;
g) Support for a single Filtering Database only. MAC Address information that is learned in one VLAN can be used in filtering decisions taken relative to all other VLANs supported by the Bridge. Bridges that support a single Filtering Database are referred to as SVL Bridges;
h) Support for multiple Filtering Databases, but only a single VLAN can be mapped onto each Filtering Database. MAC Address information that is learned in one VLAN cannot be used in filtering decisions taken relative to any other VLAN. Bridges that support this mode of operation are referred to as IVL Bridges.

6.5 VLAN classification

VLAN technology introduces the following three basic types of frame:

a) Untagged frames;
b) Priority-tagged frames; and
c) VLAN-tagged frames.

An *untagged frame* or a *priority-tagged frame* does not carry any identification of the VLAN to which it belongs. Such frames are classified as belonging to a particular VLAN based on parameters associated with the receiving Port, or, through proprietary extensions to this standard, based on the data content of the frame (e.g., MAC Address, layer 3 protocol ID, etc.).

NOTE—For the purposes of VLAN identification, priority tagged frames, which, by definition, carry no VLAN identification information, are treated the same as untagged frames.

A *VLAN-tagged frame* carries an explicit identification of the VLAN to which it belongs; i.e., it carries a tag header that carries a non-null VID. Such a frame is classified as belonging to a particular VLAN based on the value of the VID that is included in the tag header. The presence of the tag header carrying a non-null VID means that some other device, either the originator of the frame or a VLAN-aware Bridge, has mapped this frame into a VLAN and has inserted the appropriate VID. Clause 7 describes how the insertion and removal of VLAN tags is achieved.

6.6 Rules for tagging frames

For a given VLAN, all frames transmitted on a given LAN segment by a VLAN-aware Bridge shall be tagged the same way on that segment. They shall be either

a) All untagged; or
b) All VLAN tagged with the same VID.

NOTE—In other words, a Bridge can transmit untagged frames for someVLANs and VLAN-tagged frames for other VLANs on a given link, but cannot transmit both formats for the same VLAN. The single format rule expressed here only applies to the frame transmission behavior of individual VLAN-aware devices; i.e., it does not express a requirement for VLAN-aware devices to police the behavior of other devices in order to enforce a single format on a segment for all attached devices.

6.7 Spanning Tree

This standard defines a VLAN environment that operates over a single Spanning Tree. All Bridges within a Bridged LAN infrastructure participate in a single Spanning Tree, as defined by ISO/IEC 15802-3, over which multiple VLANs can coexist. As a consequence, Bridges implemented in conformance with ISO/IEC 15802-3 can be integrated into a VLAN infrastructure based on the specification contained in this standard.

NOTE 1—There are some limitations on the intermixing of ISO/IEC 15802-3 Bridges and VLAN Bridges in the same Bridged LAN, as identified in Annex E.

The primary goals of Spanning Tree are as follows:

a) Elimination of loops in a bridged infrastructure;
b) Improved scalability in a large network;
c) Provision of redundant paths, which can be activated upon failure.

There are two important items to note with respect to Spanning Tree topologies.

First, the Spanning Tree formed in a VLAN environment need not be identical to the topology of the VLAN(s). All VLANs are aligned along the Spanning Tree from which they are formed; a given VLAN is defined by a subset of the topology of the Spanning Tree upon which it operates.

Second, the topology of the VLAN is dynamic. The structure of the VLAN may change due to new devices requesting or releasing the services available via the VLAN. The dynamic nature of VLANs has the advantages of flexibility and bandwidth conservation, at the cost of network management complexity.

NOTE 2—There is a choice to be made as to how many Spanning Trees operate in a VLAN environment, and how the VLANs in that environment map to those Spanning Trees. In all cases, a given VLAN maps to a single Spanning Tree; the mapping choice to be made with multiple Spanning Trees is whether there is one Spanning Tree per VLAN, or whether many VLANs map to each Spanning Tree. Although multiple Spanning Trees offer some advantages over a single Spanning Tree in VLAN environments, this standard avoids the added complexity of defining a mapping function of VLANs to Spanning Trees by defining a VLAN environment that operates over a single Spanning Tree. It is the intent of this standard not to preclude future extensions from using multiple Spanning Trees.

In order for coexistence to be correctly maintained with Bridges implemented in conformance with ISO/IEC 15802-3, the addition or removal of tag headers by a VLAN-aware Bridge is performed only upon frames submitted to the relay function of the Bridge that are potentially to be forwarded on other Ports. Frames that carry control information that is necessary for the establishment of Spanning Tree and other aspects of the connectivity and forwarding behavior of the Bridged LAN, such as BPDUs (ISO/IEC 15802-3, Clauses 8 and 9) and GVRP PDUs (11.2), are not forwarded by a VLAN-aware Bridge. Such frames are discarded by the Forwarding Process as a result of permanently configured static entries in the Filtering Database (see 8.2, 8.3, and 8.14).

NOTE 3—GARP PDUs destined for any GARP Applications are forwarded or filtered depending upon whether the application concerned is supported by the Bridge, as specified in 8.14.

7. Support of the MAC Service in VLANs

The provisions of ISO/IEC 15802-3, Clause 6, apply to this standard, with the additions and modifications defined in this clause.

7.1 Enhanced Internal Sublayer Service provided within VLAN Bridges

The Enhanced Internal Sublayer Service (E-ISS) is derived from the Internal Sublayer Service (ISS, defined in ISO/IEC 15802-3, 6.4) by augmenting that specification with elements necessary to the operation of the tagging and untagging functions of the MAC Bridge. Within the attached end station, these elements can be considered to be either below the MAC Service boundary, and pertinent only to the operation of the service provider; or local matters not forming part of the peer-to-peer nature of the MAC Service.

Bridges that support these functions are known as *VLAN-aware Bridges* (3.17). The E-ISS defines the MAC Service provided to the relay function in VLAN-aware Bridges.

The relationships between the MAC Entity, the ISS, the E-ISS, and the MAC Relay Entity in a VLAN-aware Bridge are illustrated in Figure 7-1 and Figure 8-3.

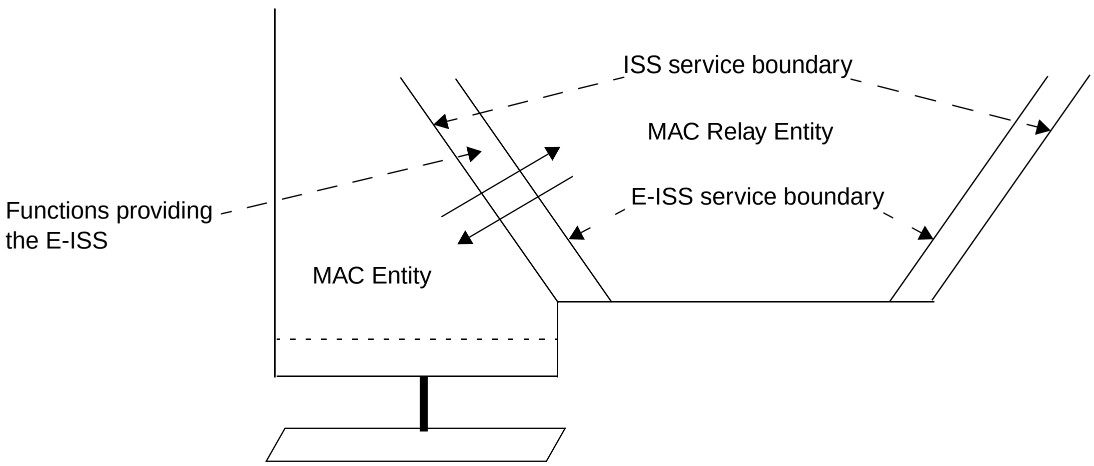


Figure 7-1—Relationships between MAC Entity, ISS, E-ISS, and MAC Relay Entity

7.1.1 E-ISS service definition

The unit-data primitives that define this service are

EM_UNITDATA.indication (
frame_type,
mac_action,
destination_address,
source_address,
mac_service_data_unit,
user_ priority,
frame_check_sequence,
canonical_format_indicator,
vlan_identifier,
rif_information (optional)
)

Each data indication primitive corresponds to the receipt of a M_UNITDATA.indication primitive from the Internal Sublayer Service, as defined in ISO/IEC 15802-3, 6.4.

The **frame_type, mac_action, destination_address, source_address, mac_service_data_unit, user_ priority,** and **frame_check_sequence** parameters are as defined for the M_UNITDATA.indication primitive of the Internal Sublayer service.

The **canonical_format_indicator** parameter indicates whether embedded MAC Addresses carried in the mac_service_data_unit parameter are in Canonical format or Non-canonical format. The value False indicates Non-canonical format. The value True indicates Canonical format.

NOTE—The meanings of the terms Canonical format and Non-canonical format are discussed in Annex F.

The **vlan_identifier** parameter carries the VLAN identifier associated with the indication.

The **rif_information** parameter is present if a tag header was present in the indication, and if that tag header contained a Routing Information Field (RIF). Its value is equal to the value of the RIF.

```
EM_UNITDATA.request     (
                        frame_type,
                        mac_ action,
                        destination_address,
                        source_address,
                        mac_service_data_unit,
                        user_priority,
                        access_priority,
                        frame_check_sequence,
                        canonical_format_indicator,
                        vlan_classification,
                        rif_information (optional),
                        include_tag
                        )
```

A data request primitive is invoked in order to generate a M_UNITDATA.request primitive, as defined in the Internal Sublayer Service, ISO/IEC 15802-3, 6.4.

The **frame_type, mac_action, destination_address, source_address, mac_service_data_unit, user_priority, access_priority,** and **frame_check_sequence** parameters are as defined for the M_UNITDATA.request primitive of the Internal Sublayer service.

The definition of the **canonical_format_indicator** parameter is as defined for the EM_UNITDATA.indication.

The **vlan_classification** parameter carries the VLAN classification assigned to the frame by the ingress rules (8.6).

The **rif_information parameter**, if present, carries the value of any RIF information to be associated with the request.

The **include_tag** parameter carries a Boolean value. True indicates to the service provider that the mac_service_data_unit parameter of the data request shall include a tag header (9.3). False indicates that a tag header shall not be included.

7.1.2 Support of the E-ISS in VLAN-aware Bridges

7.1.2.1 Data indication primitives

On receipt of a data indication from the Internal Sublayer Service, an EM_UNITDATA.indication primitive is invoked, with parameter values as follows:

The **frame_type, mac_action, destination_address, source_address,** and **frame_check_sequence** parameters carry values equal to the corresponding parameters in the received data indication.

NOTE 1—The **mac_action** parameter only ever takes the value request_with_no_response for frames relayed by the Bridge. The **frame_check_sequence** parameter of the data indication carries the FCS value contained in the received frame. The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signalled in the E-ISS by an unspecified value in the frame_check_sequence parameter of the data request primitive. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in ISO/IEC 15802-3, Annex G.

The value of the **mac_service_data_unit** parameter is determined as follows:

a) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value used is equal to the value of the received mac_service_data_unit following removal of the tag header. Otherwise;
b) The value used is equal to the value of the received mac_service_data_unit.

The value of the **user_priority** parameter is determined as follows:

c) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value contained in the user_priority field of the tag header is used. Otherwise;
d) The value of the received user_ priority parameter, regenerated as defined in 8.5.1 and ISO/IEC 15802-3, 6.4, is used.

The value of the **canonical_format_indicator** parameter is determined as follows:

e) If the received mac_service_data_unit parameter contained a tag header (9.3), then the value(s) contained in the Canonical Format Indicator (CFI) (and Non-Canonical Format Indicator [NCFI], if present) field(s) of the tag header are used to determine this parameter value, in accordance with the definition of the CFI and NCFI field(s) in Clause 9. Otherwise;
f) If the MAC entity that received the data indication was an ISO/IEC 8802-5 Token Ring MAC, then the parameter carries the value False. Otherwise;
g) The parameter carries the value True.

The value of the **vlan_identifier** parameter is determined as follows:

h) If the initial octets of the received mac_service_data_unit parameter contained a tag header (9.3), then the value contained in the VID field of the tag header is used. Otherwise;
i) A value equal to the null VLAN ID (as defined in Table 9-2) is used.

The value of the **rif_information** parameter is determined as follows:

j) If the initial octets of the received mac_service_data_unit parameter contained a tag header (9.3), and that tag header contained a RIF field in which one or more route descriptors were present, then the value contained in the RIF field is used. Otherwise;
k) The parameter is not present.

NOTE 2—This field can be present only in tag headers received using the 802.3/Ethernet or transparent FDDI MAC methods. The presence of one or more route descriptors indicates that there is source-routing information present in the received frame.

7.1.2.2 Data request primitives

On invocation of a data request primitive by a user of the E-ISS, an M-UNITDATA.request primitive is invoked, with parameter values as follows:

The **frame_type, mac_action, destination_address, source_address, user_priority,** and **access_priority** parameters carry values equal to the corresponding parameters in the received data request.

If the value of the **include_tag** parameter is False, the value of the mac_service_data_unit parameter is determined as follows:

a) If the destination MAC method is the same as the MAC method on which the corresponding data indication was received, then the value used is equal to the value of the mac_service_data_unit parameter received in the data request. Otherwise;
b) The value used is equal to the value of the mac_service_data_unit parameter received in the data request, modified, if necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.
c) If the canonical_format_indicator parameter indicates that the mac_service_data_unit may contain embedded MAC Addresses in a format inappropriate to the destination MAC method, then the Bridge shall either
    1) Convert any embedded MAC Addresses in the mac_service_data_unit to the format appropriate to the destination MAC method; or
    2) Discard the EISS data request without issuing a corresponding ISS data request.

If the value of the **include_tag parameter** is True, then a tag header, formatted as necessary for the destination MAC method, is inserted as the first N octets of the mac_service_data_unit parameter. The values of the user_priority, canonical_format_indicator, vlan_classification, and rif_information (if present) parameters are used to determine the contents of the tag header, in accordance with the structure defined in 9.2 and 9.3. The value inserted after the tag header is determined as follows:

d) If the destination MAC method is the same as the MAC method on which the corresponding data indication was received, then the value used is equal to the value of the mac_service_data_unit parameter received in the data request. Otherwise;
e) The value used is equal to the value of the mac_service_data_unit parameter received in the data request, modified, if necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.

The value of the **frame_check_sequence** parameter is determined as follows:

f) If the frame_check_sequence parameter received in the data request is either unspecified or still carries a valid value, then that value is used. Otherwise;
g) The value used is either derived from the received FCS information by modification to take account of the conditions that have caused it to become invalid, or the unspecified value is used.

NOTE—The original FCS associated with a frame is invalidated if there are changes to any fields of the frame, if fields are added or removed, or if bit ordering or other aspects of the frame encoding have changed. An invalid FCS is signalled in the E-ISS by an unspecified value in the frame_check_sequence parameter of the data request primitive. This signals the need for the FCS to be regenerated according to the normal procedures for the transmitting MAC. The options for regenerating the FCS under these circumstances are discussed in ISO/IEC 15802-3, Annex G.

7.2 Support of the Internal Sublayer Service by IEEE Std 802.3 (CSMA/CD)

In addition to the provisions of ISO/IEC 15802-3, 6.5.1, on receipt of an M_UNITDATA.request primitive that represents a tagged frame, the implementation is permitted to adopt either of the following approaches with regard to the operation of Transmit Data Encapsulation for frames whose length would, using the procedure as described, be less than 68 octets:

a) Use the procedure as described in ISO/IEC 15802-3, 6.5.1. This can result in tagged frames of less than 68 octets (but at least 64 octets) being transmitted; or
b) Include additional octets before the FCS field in order for the transmitted frame length for such frames to be 68 octets. This results in a minimum tagged frame length of 68 octets.

When a tagged frame of less than 68 octets in length is received on a CSMA/CD LAN segment, and is forwarded as an untagged frame, the provisions of ISO/IEC 15802-3, 6.5.1, result in additional octets being included before the FCS field on transmission in order that the transmitted frame length meets the minimum frame size requirements of IEEE Std 802.3, 1998 Edition, 3.2.7.

8. Principles of operation

This clause establishes the principles of operation of a VLAN-aware Bridge, by reference to a model of that operation, as follows:

a) Explains the principal elements of Bridge operation and lists the functions that support these.
b) Establishes an architectural model for a Bridge that governs the provision of these functions.
c) Provides a model of the operation of a Bridge in terms of the processes and entities that support the functions.
d) Details the addressing requirements in a Bridged LAN and specifies the addressing of entities in a Bridge.

The provisions of this clause replace the provisions of ISO/IEC 15802-3, Clause 7, in a VLAN-aware Bridge.

8.1 Bridge operation

The principal elements of Bridge operation are

a) Relay and filtering of frames.
b) Maintenance of the information required to make frame filtering and relaying decisions.
c) Management of the above.

8.1.1 Relay

A MAC Bridge relays individual MAC user data frames between the separate MACs of the Bridged LANs connected to its Ports. The order of frames shall be preserved as defined in 8.7.3.

The functions that support the relaying of frames and maintain the Quality of Service supported by the Bridge are

a) Frame reception.
b) Discard on received frame in error (ISO/IEC 15802-3, 6.3.2).
c) Frame discard if the frame_type is not user_data_frame, or if its mac_action parameter is not request_with_no_response (8.5, ISO/IEC 15802-3, 6.4).
d) Regeneration of user priority, if required (ISO/IEC 15802-3, 6.4).
e) Frame discard following the application of filtering information.
f) Frame discard on transmittable service data unit size exceeded (ISO/IEC 15802-3, 6.3.8).
g) Forwarding of received frames to other Bridge Ports.
h) Selection of traffic class, following the application of filtering information.
i) Queuing of frames by traffic class.
j) Frame discard to ensure that a maximum bridge transit delay is not exceeded (ISO/IEC 15802-3, 6.3.6).
k) Selection of queued frames for transmission.
l) Selection of outbound access priority (ISO/IEC 15802-3, 6.3.9).
m) Mapping of service data units and recalculation of Frame Check Sequence, if required (8.7.6, ISO/IEC 15802-3, 6.3.7).
n) Frame transmission.

8.1.2 Filtering and relaying information

A Bridge filters frames, i.e., does not relay frames received by a Bridge Port to other Ports on that Bridge, in order to prevent the duplication of frames (ISO/IEC 15802-3, 6.3.4). The function that supports the use and maintenance of information for this purpose is

a) Calculation and configuration of Bridged LAN topology.

A Bridge also filters frames in order to reduce traffic in parts of the Bridged LAN that do not lie in the path between the source and destination of that traffic. The functions that support the use and maintenance of information for this purpose are

b) Permanent configuration of reserved addresses.
c) Explicit configuration of static filtering information.
d) Automatic learning of dynamic filtering information for unicast destination addresses through observation of source addresses of Bridged LAN traffic.
e) Ageing out of dynamic filtering information that has been learned.
f) Automatic addition and removal of dynamic filtering information as a result of GMRP protocol exchanges.

A Bridge classifies frames into traffic classes in order to expedite transmission of frames generated by critical or time-sensitive services. The function that supports the use and maintenance of information for this purpose is

g) Explicit configuration of traffic class information associated with the Ports of the Bridge.

A Bridge classifies untagged frames and priority-tagged frames as belonging to a particular VLAN in accordance with the *ingress rules* defined in 8.6. The function that supports the use and maintenance of information for this purpose is

h) Explicit configuration of the Port VID (PVID, 8.4.4) associated with each Port of the Bridge.

A Bridge may filter frames in order to prevent the injection of untagged and priority-tagged frames on a Port on which the reception of untagged and priority-tagged frames is disallowed. The function that supports the use and maintenance of information for this purpose is

i) Explicit configuration of the Acceptable Frame Types parameter (8.4.3) associated with each Port of the Bridge.

A Bridge may filter frames in order to prevent the injection of traffic for a given VLAN on a Port on which that VLAN is disallowed. The function that supports the use and maintenance of information for this purpose is

j) Explicit configuration of the Enable Ingress Filtering parameter (8.4.5) associated with each Port of the Bridge.

A Bridge filters frames in order to confine traffic destined for a given VLAN to LAN segments that form a path from the source of the traffic to recipients that are members of that VLAN. The functions that support the use and maintenance of information for this purpose are

k) Automatic configuration of Dynamic VLAN Registration Entries by means of GVRP (8.11.5 and 11.2);
l) Explicit configuration of management controls associated with the operation of GVRP by means of Static VLAN Registration Entries (8.11.2 and 11.2);
m) Automatic learning of MAC Addresses in associated VLANs through the observation of network traffic (8.10).

A Bridge adds and removes tag headers (9.3) from frames, and performs the associated frame translations that may be required, in accordance with the *egress rules* (8.8). The function that supports the use and maintenance of information for this purpose is

n) Explicit configuration of tagging requirements on egress for each Port (8.11.2 and 8.11.9).

8.1.3 Bridge management

The functions that support Bridge Management control and monitor the provision of the above functions. They are specified in Clause 12.

8.2 Bridge architecture

8.2.1 Architectural model of a Bridge

Figure 8-1 gives an example of the physical topology of a Bridged LAN. The component LANs are interconnected by means of MAC Bridges; each Port of a MAC Bridge connects to a single LAN. Figure 8-2 illustrates a Bridge with two Ports, and Figure 8-3 illustrates the architecture of such a Bridge.

A Bridge is modeled as consisting of

a) A MAC Relay Entity that interconnects the Bridge's Ports;
b) At least two Ports;
c) Higher layer entities, including at least a Bridge Protocol Entity.

8.2.2 MAC Relay Entity

The MAC Relay Entity handles the MAC method independent functions of relaying frames between Bridge Ports, filtering frames, and learning filtering information. It uses the Internal Sublayer Service provided by the separate MAC Entities for each Port. (The Internal Sublayer Service and its support are described in ISO/IEC 15802-3, 6.4 and 6.5.) Frames are relayed between Ports attached to different LANs.

8.2.3 Ports

Each Bridge Port transmits and receives frames to and from the LAN to which it is attached. An individual MAC Entity permanently associated with the Port provides the Internal Sublayer Service used for frame transmission and reception. The MAC Entity handles all the MAC method dependent functions (MAC protocol and procedures) as specified in the relevant standard for that IEEE 802 LAN MAC technology.

8.2.4 Higher Layer Entities

The Bridge Protocol Entity handles calculation and configuration of Bridged LAN topology.

The Bridge Protocol Entity and other higher layer protocol users, such as Bridge Management (8.1.3) and GARP application entities including GARP Participants (ISO/IEC 15802-3, Clause 12), make use of Logical Link Control procedures. These procedures are provided separately for each Port, and use the MAC Service provided by the individual MAC Entities.

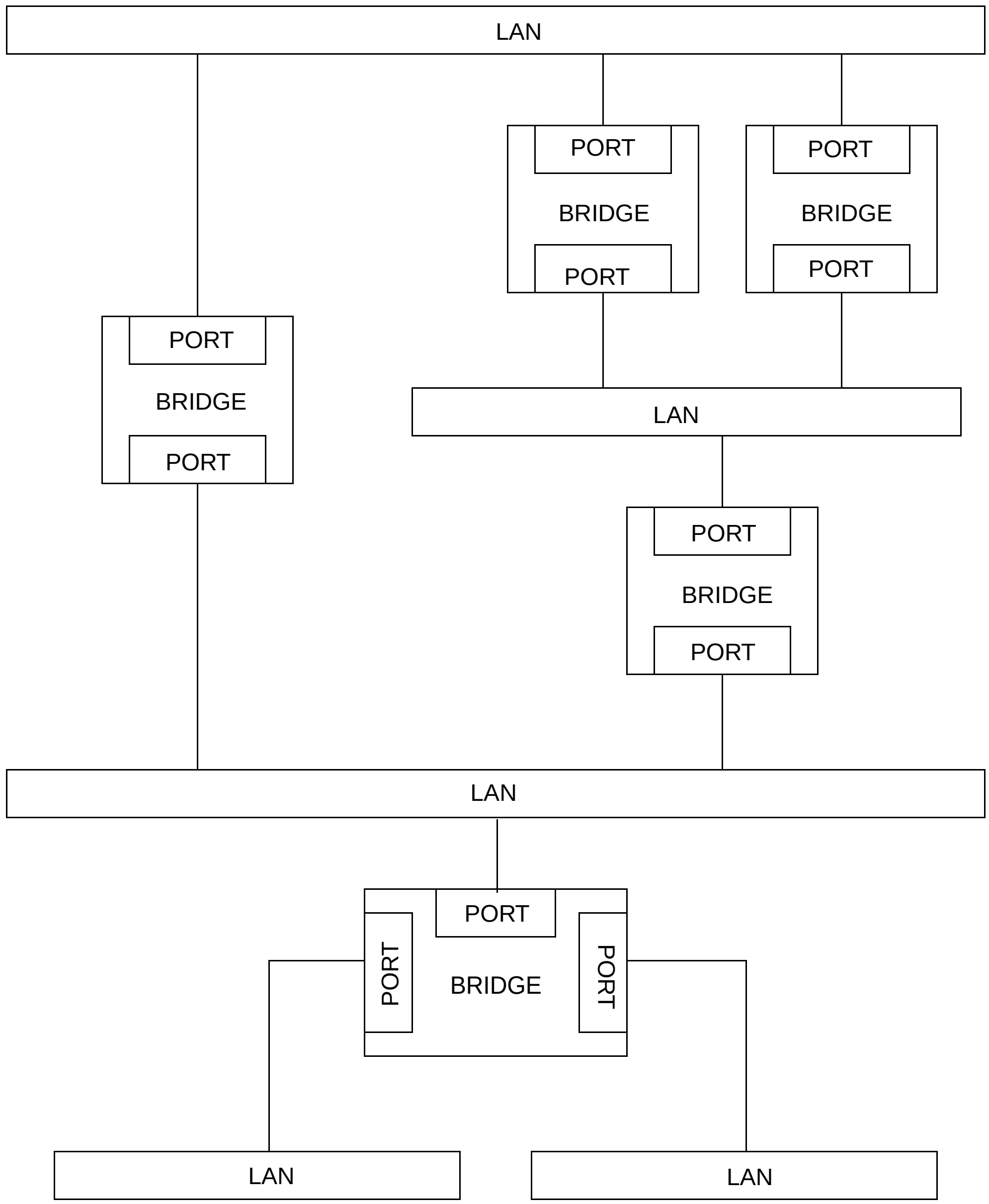


Figure 8-1—Example of a Bridged LAN

8.3 Model of operation

The model of operation is simply a basis for describing the functionality of the MAC Bridge. It is in no way intended to constrain real implementations of a MAC Bridge; these may adopt any internal model of operation compatible with the externally visible behavior that this standard specifies. Conformance of equipment to this standard is purely in respect of observable protocol.

Subclauses 8.5 and 8.9 specify the MAC Relay Entity's use of the Internal Sublayer Service. State information associated with each Port governs the Port's participation in the Bridged LAN. (Port States are specified in detail in ISO/IEC 15802-3, 8.4.)

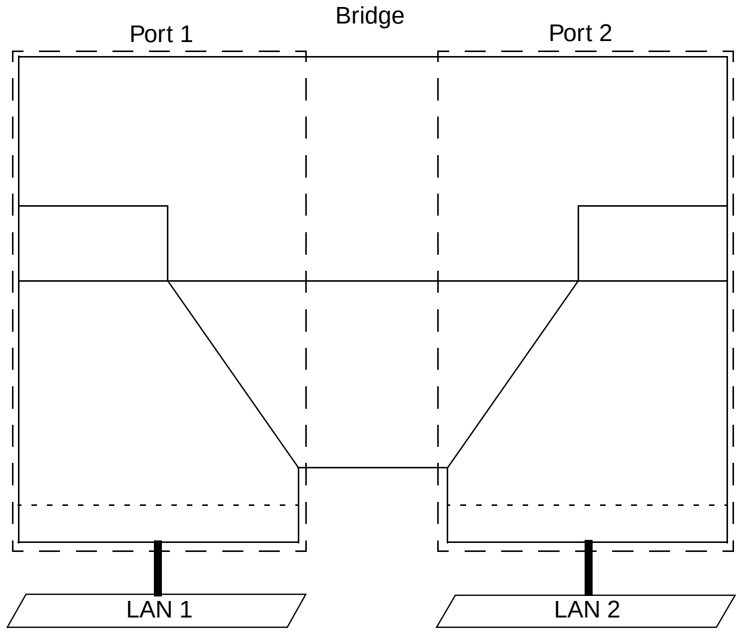


Figure 8-2—Bridge ports

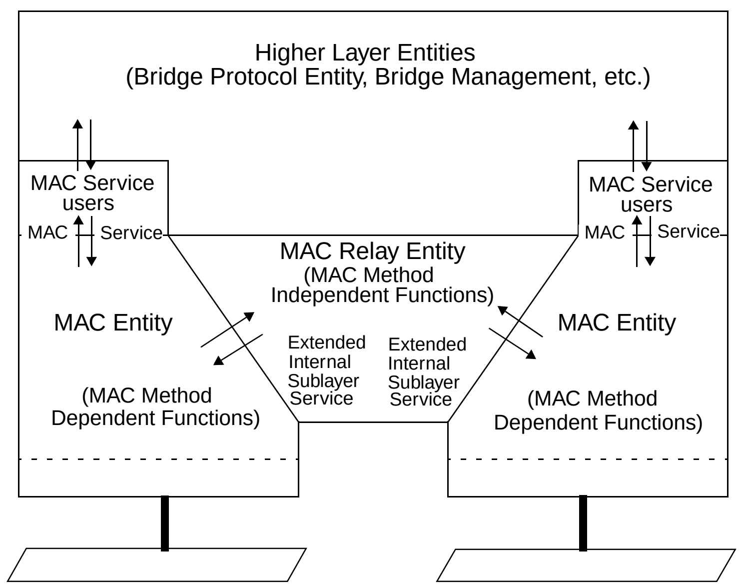


Figure 8-3—VLAN Bridge architecture

Frames are accepted for transmission and delivered on reception to and from processes and entities that model the operation of the MAC Relay Entity in a Bridge. These are

a) The ingress rules (8.6), which classify received frames according to their VLAN membership, may filter frames based on the absence of a VID in the received frame (8.4.3), and may filter frames based on the frame's VLAN identifier (8.4.5);
b) The Forwarding Process (8.7), which forwards received frames that are to be relayed to other Bridge Ports, filtering frames on the basis of information contained in the Filtering Database (8.11) and on the state of the Bridge Ports (8.4);
c) The egress rules (8.8), which determine, for a given VLAN, through which Ports frames may be transmitted, and in what format;
d) The Learning Process (8.10), which, by observing the source addresses and VIDs of frames classified by the ingress rules, updates the Filtering Database (8.11), conditionally on the Port state (8.4);
e) The Filtering Database (8.11), which holds filtering information and supports queries by the Forwarding Process as to whether frames with given values of the destination MAC Address field and VID should be forwarded to a given Port.

Each Bridge Port also functions as an end station providing the MAC Service to LLC, which in turn supports operation of the Bridge Protocol Entity (8.12) and of other possible users of LLC, such as protocols providing Bridge Management (8.13).

Each Bridge Port shall support the operation of LLC Type 1 procedures in order to support the operation of the Bridge Protocol Entity. Bridge Ports may support other types of LLC procedures, which may be used by other protocols.

Figure 8-4 illustrates a single instance of frame relay between the Ports of a Bridge with two Ports.

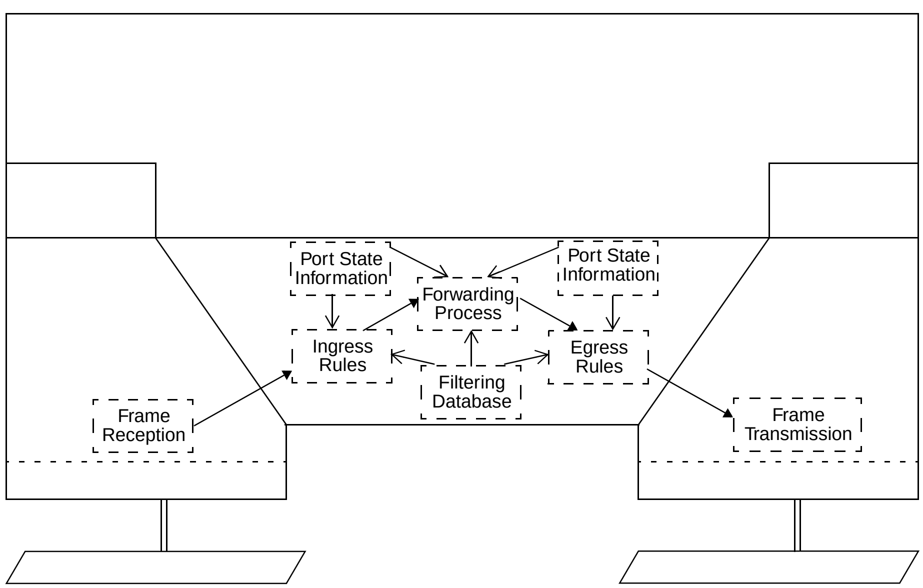


Figure 8-4—Relaying MAC frames

Figure 8-5 illustrates the inclusion of information carried by a single frame, received on one of the Ports of a Bridge with two Ports, in the Filtering Database.

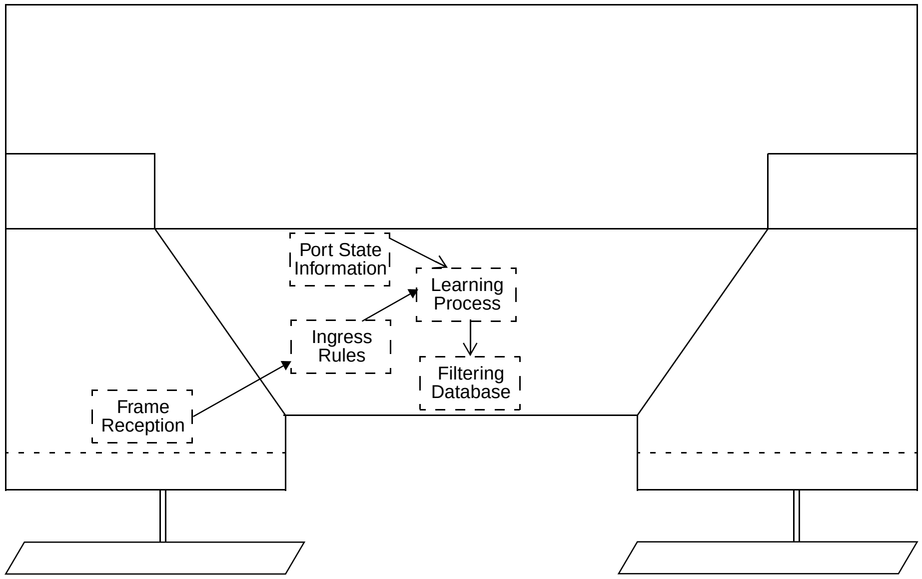


Figure 8-5—Observation of network traffic

Figure 8-6 illustrates the reception and transmission of Bridge Protocol Data Units by the Bridge Protocol Entity.

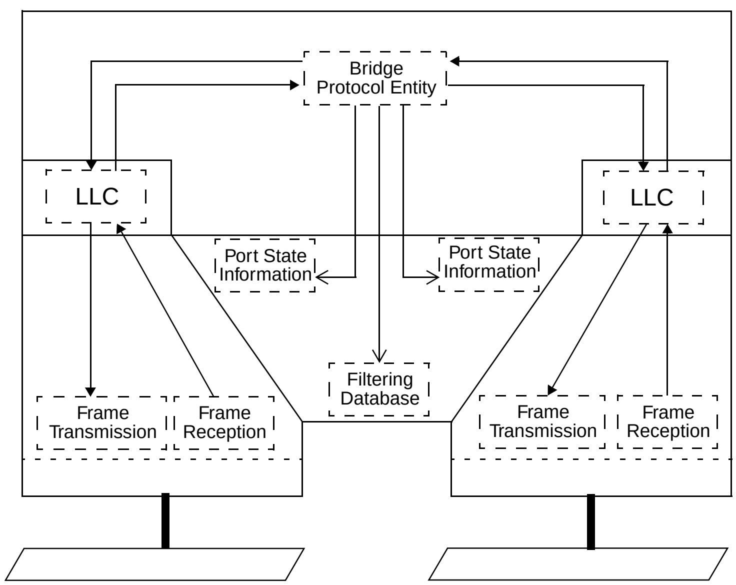


Figure 8-6—Operation of inter-bridge protocol

Figure 8-7 illustrates the reception and transmission of GARP Protocol Data Units by a GARP Protocol Entity (8.12).

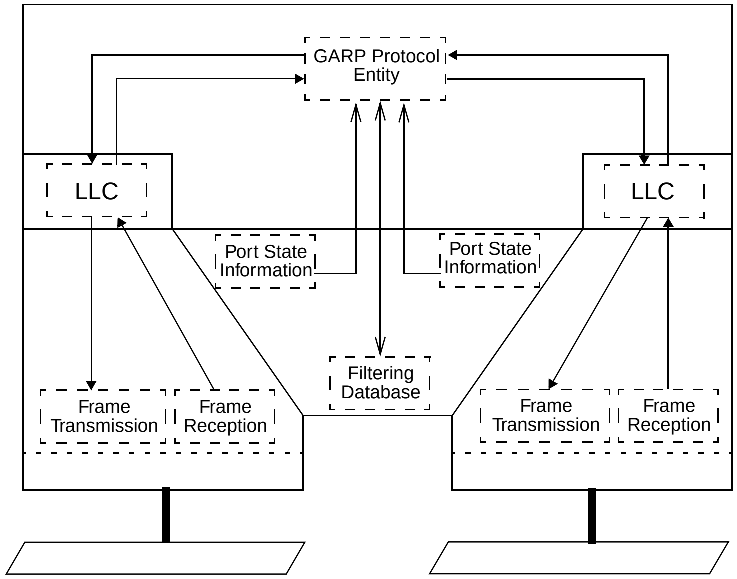


Figure 8-7—Operation of the GARP protocol

8.4 Port States, Port parameters, Active Ports, and the active topology

8.4.1 Forwarding states

State information associated with each Bridge Port governs whether or not it participates in relaying MAC frames. A Port can be disabled by management, in which case it plays no part in the operation of the Bridged LAN; a Port that is not disabled can be dynamically excluded from participation in frame relaying by operation of the Spanning Tree algorithm. If neither of these applies to a Port, it is described as *forwarding*.

The *active topology* of a Bridged LAN at any time is the set of communication paths formed by interconnecting the LANs and Bridges by the forwarding Ports. The function of the distributed Spanning Tree algorithm (ISO/IEC 15802-3, Clause 8) is to construct an active topology that is simply connected relative to communication between any given pair of MAC Addresses used to address end stations on the LANs.

Figure 8-6 illustrates the operation of the Bridge Protocol Entity, which operates the Spanning Tree algorithm and its related protocols, and its modification of Port state information as part of determining the active topology of the Bridged LAN. The Port states associated with the determination of the active topology are specified in detail in ISO/IEC 15802-3, 8.4.

Figure 8-4 illustrates the Forwarding Process's use of Port state information: first, for a Port receiving a frame, in order to determine whether the received frame is to be relayed through any other Ports; and second, for another Port in order to determine whether the relayed frame is to be forwarded through that particular Port.

8.4.2 Learning states

The incorporation of end station location information in the Filtering Database by the Learning Process also depends on the active topology. If information associated with frames received on a Port is to be incorpo-

rated in the Filtering Database by the Learning Process, then the Port is described as being in a learning state; otherwise, it is in a non-learning state. Figure 8-5 illustrates the use of the Port state information for a Port receiving a frame, by the Learning Process, in order to determine whether the station location information is to be incorporated in the Filtering Database.

8.4.3 Acceptable Frame Types

Associated with each Port of a VLAN Bridge is an Acceptable Frame Types parameter that controls the reception of VLAN-tagged and non VLAN-tagged frames on that Port. Valid values for this parameter are

a) *Admit Only VLAN-tagged frames*;
b) *Admit All Frames*.

If this parameter is set to *Admit Only VLAN-tagged frames*, any frames received on that Port that carry no VID (i.e., untagged frames or priority-tagged frames) are discarded by the ingress rules (8.6).

Frames that are not discarded as a result of this parameter value are classified and processed according to the ingress rules that apply to that Port.

Each Port of the Bridge shall support at least one of these values, and may support both. Where both values are supported,

c) The implementation shall support the ability to configure the value of the parameter by means of the management operations defined in Clause 12; and
d) The default value of the parameter shall be *Admit All Frames*.

8.4.4 Port VLAN identifier

In Port-based VLAN classification within a Bridge, the VID associated with an untagged or priority-tagged frame (i.e., a frame with no tag header, or a frame with a tag header that carries the null VLAN ID) is determined, based on the Port of arrival of the frame into the Bridge, as described in 8.6. This classification mechanism requires the association of a specific VLAN ID, the *Port VLAN Identifier*, or *PVID*, with each of the Bridge's Ports.

The PVID for a given Port provides the VID for untagged and priority-tagged frames received through that Port. The PVID for each Port shall contain a valid VID value, and shall not contain the value of the null VLAN ID (Table 9-2).

NOTE—This rule ensures that the process of ingress classification of frames always associates a non-null VID with each received frame. As a consequence, a VLAN-aware Bridge can never transmit priority-tagged frames; all frames transmitted are either untagged or carry a non-null VID in their tag header.

The PVID value may be configured by management, if management operations are supported by the implementation. If no PVID value has been explicitly configured for a Port, the PVID shall assume the value of the default PVID defined in Table 9-2.

8.4.5 Enable Ingress Filtering

An Enable Ingress Filtering parameter is associated with each Port. If the Enable Ingress Filtering parameter for a given Port is set, the ingress rules (8.6) shall discard any frame received on that Port whose VLAN classification does not include that Port in its Member set (8.11.9). If the parameter is reset for that Port, the ingress rules shall not discard frames received on that Port on the basis of their VLAN classification.

The default value for this parameter is reset, i.e., Disable Ingress Filtering, for all Ports. The value of this parameter may be configured by means of the management operations defined in Clause 12, if management operations are supported by the implementation. If the implementation supports the ability to enable Ingress Filtering on any Port, then it shall also support the ability to disable Ingress Filtering on those Ports.

8.5 Frame reception

The individual MAC Entity associated with each Bridge Port examines all frames received on the LAN to which it is attached.

All error-free received frames give rise to EM_UNITDATA indication primitives, which shall be handled as follows.

NOTE—A frame that is in error, as defined by the relevant MAC specification, is discarded by the MAC Entity without giving rise to any EM_UNITDATA indication: see 7.2 and ISO/IEC 15802-3, 6.4.

Frames with EM_UNITDATA.indication primitive frame_type and mac_action parameter values of user_data_frame and request_with_no_response, respectively (7.2 and ISO/IEC 15802-3, 6.4), shall be submitted to the ingress rules (8.6).

Frames with other values of frame_type and mac_action parameters, (e.g., request_with_response and response frames), shall not be submitted to the ingress rules (8.6).

Frames with a frame_type of user_data_frame and addressed to the Bridge Port as an end station shall be submitted to the MAC Service user. Such frames carry either the individual MAC Address of the Port or a group address associated with the Port (8.14) in the destination address field. Frames submitted to the MAC Service user can also be submitted to the ingress rules (8.6), as specified above.

Frames addressed to a Bridge Port as an end station, and relayed to that Bridge Port from other Bridge Ports in the same Bridge by the Forwarding Process, shall also be submitted to the MAC Service user.

NOTE—The consequence of the above is that frames “relayed to that Bridge Port” are both submitted to that Port’s MAC Service user and transmitted on the LAN to which that Port is attached (see 8.14.7).

No other frames shall be submitted to the MAC Service user.

8.5.1 Regenerating user priority

The user_priority of received frames is regenerated using priority information contained in the frame and the User Priority Regeneration Table for the reception Port. For each reception Port, the User Priority Regeneration Table has eight entries, corresponding to the eight possible values of user_priority (0 through 7). Each entry specifies, for the given value of received user_priority, the corresponding Regenerated user_priority value.

NOTE 1—IEEE 802 LAN technologies signal a maximum of eight user_priority values. Annex H.2 of ISO/IEC 15802-3 contains further explanation of the use of user_priority values and how they map to traffic classes.

Table 8-1 defines the default values of Regenerated user_priority for the eight possible values of the user_priority parameter received in a data indication; these values shall be used as the initial values of the corresponding entries of the User Priority Regeneration Table for each Port.

Optionally, the ability to modify the values in the User Priority Regeneration Table by management means may be supported, as described in Clause 12. If this capability is provided, the value of the table entries may

be independently settable for each reception Port and for each value of received user_priority, and the Bridge may have the capability to use the full range of values in the parameter ranges specified in Table 8-1.

NOTE 2—It is important to ensure that the regeneration and mapping of user priority within the Bridge is consistent with the end-to-end significance attached to that user priority in the Bridged LAN. Within a given Bridge, the values chosen for the User Priority Regeneration Table for a given Port should be consistent with the priority to be associated with traffic received through that Port across the rest of the Bridged LAN, and should generate appropriate access priority values for each transmission MAC method. The user priority value regenerated via the User Priority Regeneration Table on reception is used:

— Via the traffic class table (8.7.3) to determine the traffic class for a given outbound Port, and
— Via fixed, MAC method-specific mappings (8.7.5) to determine the access priority that will be used for a given outbound MAC method.

Table 8-1 shows the default values for the regeneration of user priority. Table 8-2 shows the default values for the traffic class table, for all possible numbers of supported traffic classes. Table 8-3 shows the fixed mappings from user priority to access priority that are required for different outbound MAC methods.

Table 8-1—User priority regeneration

User priority	Default regenerated user priority	Range
0	0	0–7
1	1	0–7
2	2	0–7
3	3	0–7
4	4	0–7
5	5	0-7
6	6	0–7
7	7	0–7

8.6 The ingress rules

If the vlan_identifier parameter carried in a received data indication is equal to the null VLAN ID (Table 9-2) and the Acceptable Frame Types parameter (8.4.3) for the Port through which the frame was received is set to the value *Admit Only VLAN-tagged frames*, then the frame shall be discarded.

Each frame received by a VLAN Bridge shall be classified as belonging to exactly one VLAN by associating a VID value with the received frame. The classification is achieved as follows:

a) If the vlan_identifier parameter carried in a received data indication is the null VLAN ID (Table 9-2), then
   1) If the implementation supports further VLAN classification rules in addition to Port-based classification (D.2.2), and if the application of these rules associates a non-null VID value with the frame, then that VID value is used.

2) If the implementation supports only Port-based classification, or if any additional classification rules supported are unable to associate a non-null VID with the frame, then the PVID value associated with the Port through which the frame was received is used (8.4.4).

b) If the vlan_identifier parameter carried in a received data indication is not the null VLAN ID (Table 9-2), then the vlan_identifier parameter value is used.

NOTE 1—As defined in 7.1.2, the vlan_identifier parameter carries the null VLAN ID if the frame was not VLAN-tagged. There are two cases; either the frame was untagged, or the frame was tagged and the tag header carried a VID value equal to the null VLAN ID (i.e., a priority-tagged frame).
NOTE 2—VIDs of value FFF cannot be configured in any Filtering Database entry (see Table 9-2). Consequently, any incoming frame whose VLAN classification is FFF will be discarded by the Forwarding Process.

The VID value thus identified, known as the *VLAN classification* of the frame, is used as the value of the vlan_classification parameter of any corresponding data request primitives.

If the Enable Ingress Filtering parameter (8.4.5) for the Port through which the frame was received is set, and if the Port is not in the Member set (8.11.9) for the frame's VLAN classification, then the frame is discarded.

All frames that are not discarded as a result of the application of the ingress rules are submitted to the Forwarding Process and to the Learning Process. All frames that are discarded as a result of the application of the ingress rules are not submitted either to the Forwarding Process or to the Learning Process.

8.7 The Forwarding Process

Frames submitted to the Forwarding Process after being received at any given Bridge Port (8.5) shall be forwarded through the other Bridge Ports subject to the constituent functions of the Forwarding Process. These functions enforce topology restrictions (8.7.1), use Filtering Database information to filter frames (8.7.2), queue frames (8.7.3), select queued frames for transmission (8.7.4), map priorities (8.7.5), and recalculate FCS if required (8.7.6).

The Forwarding Process functions are described in 8.7.1–8.7.6 in terms of the action taken for a given frame received on a given Port (termed "the reception Port"). The frame can be forwarded for transmission on some Ports (termed "transmission Ports"), and is discarded without being transmitted at the other Ports.

NOTE—The model of operation of the Forwarding Process described in this standard is limited to the operation of the relay function of the MAC Bridge, and does not take into consideration what may occur in real implementations once frames are passed to the MAC for transmission. In some MAC implementations, and under some traffic conditions, a degree of indeterminacy may be introduced between the modeled description of the process of passing selected frames to the MAC for transmission and the actual sequence of frames as visible on the LAN medium itself. Examples can be found in the handling of access_priority in Token-Passing Bus MACs, or in the effect of different values for Token Holding Time in FDDI LANs. Such indeterminacy could result in apparent violation of the queuing/de-queueing and prioritizing rules described for the Forwarding Process, when observing traffic on the medium. As a consequence, in some implementations of this standard, it may prove to be impossible to test conformance to the standard simply by relating observed LAN traffic to the described model of the forwarding process; conformance tests would have to allow for the (permissible) behavior of the MAC implementations as well.

Figure 8-4 illustrates the operation of the Forwarding Process in a single instance of frame relay between the Ports of a Bridge with two Ports. Figure 8-8 illustrates the detailed operation of the Forwarding Process.

8.7.1 Enforcing topology restriction

Each Port is selected as a potential transmission Port if, and only if

a) The Port on which the frame was received was in a forwarding state (ISO/IEC 15802-3, 8.4), and
b) The Port considered for transmission is in a forwarding state, and

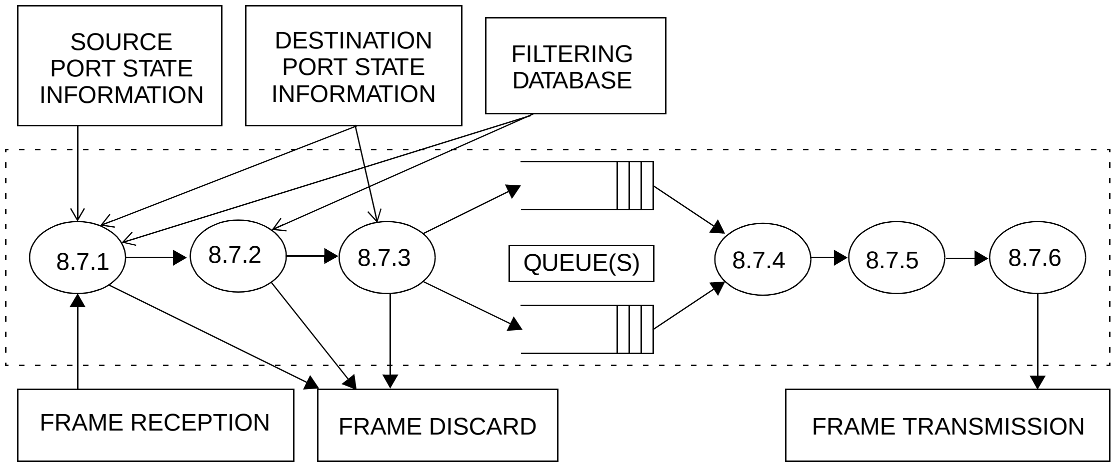


Figure 8-8—Illustration of the detailed operation of the Forwarding Process

c) The Port considered for transmission is not the same as the Port on which the frame was received, and
d) The size of the mac_service_data_unit conveyed by the frame does not exceed the maximum size of mac_service_data_unit supported by the LAN to which the Port considered for transmission is attached.

For each Port not selected as a potential transmission Port the frame shall be discarded.

8.7.2 Filtering frames

Filtering decisions are taken by the Forwarding Process on the basis of

a) The destination MAC Address carried in a received frame;
b) The VID associated with the received frame;
c) The information contained in the Filtering Database for that MAC Address and VID;
d) The default Group filtering behavior for the potential transmission Port (8.11.6).

For each potential transmission Port selected as in 8.7.1, the frame shall be forwarded, or discarded (i.e., filtered), on the basis of this information, in accordance with the definition of the Filtering Database entry types (8.11.1, 8.11.3, and 8.11.4). The required forwarding and filtering behavior is summarized in 8.11.6, 8.11.8, Table 8-5, Table 8-6, and Table 8-7.

8.7.3 Queuing frames

The Forwarding Process provides storage for queued frames, awaiting an opportunity to submit these for transmission to the individual MAC Entities associated with each Bridge Port. The order of frames received on the same Bridge Port shall be preserved for

a) Unicast frames with a given user_priority (regenerated as defined in 8.5.1) for a given combination of destination_address and source_address;
b) Group-addressed frames with a given user_priority (regenerated as defined in 8.5.1) for a given destination_address.

The Forwarding Process may provide more than one transmission queue for a given Bridge Port. Frames are assigned to storage queue(s) on the basis of their user_priority using a traffic class table that is part of the

state information associated with each Port. The table indicates, for each possible value of user_priority, the corresponding value of traffic class that shall be assigned. Values of user_priority range from 0 through 7. Queues correspond one-to-one with traffic classes.

NOTE 1—Annex H.2 of ISO/IEC 15802-3 contains further explanation of the use of user_priority values and how they map to traffic classes.

For management purposes, up to eight traffic classes are supported by the traffic class tables in order to allow for separate queues for each level of user_priority. Traffic classes are numbered 0 through N-1, where N is the number of traffic classes associated with a given outbound Port. Management of traffic class information is optional. Traffic class 0 corresponds to non-expedited traffic; non-zero traffic classes are expedited classes of traffic.

NOTE 2—In a given Bridge, it is permissible to implement different numbers of traffic classes for each Port. Ports associated with MAC methods that support a single transmission priority, such as CSMA/CD, can support more than one traffic class.

Where the Forwarding Process does not support expedited classes of traffic for a given Port, in other words, where there is a single traffic class associated with the Port, all values of user_priority map to traffic class 0. In bridges which support expedited traffic, the recommended mapping of user_priority to traffic class, for the number of traffic classes implemented, is as shown in Table 8-2. Each entry in the body of the table is the traffic class assigned to traffic with a given user_priority, for a given number of available traffic classes.

Table 8-2—Recommended user priority to traffic class mappings

		Number of Available Traffic Classes							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
User Priority	**0 (Default)**	0	0	0	1	1	1	1	2
	1	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	1
	3	0	0	0	1	1	2	2	3
	4	0	1	1	2	2	3	3	4
	5	0	1	1	2	3	4	4	5
	6	0	1	2	3	4	5	5	6
	7	0	1	2	3	4	5	6	7

NOTE—The rationale behind the choice of values shown in this table is discussed in Annex H.2 of ISO/IEC 15802-3. A consequence of the mapping shown is that frames carrying the default user priority are given preferential treatment relative to user priority 1 and 2 in Bridges that implement four or more Traffic Classes.

A frame queued by the Forwarding Process for transmission on a Port shall be removed from that queue on submission to the individual MAC Entity for that Port. No further attempt shall be made to transmit the frame on that Port even if the transmission is known to have failed.

A frame queued by the Forwarding Process for transmission on a Port can be removed from that queue, and not subsequently transmitted, if the time for which buffering is guaranteed has been exceeded for that frame.

A frame queued for transmission on a Port shall be removed from that queue if that is necessary to ensure that the maximum bridge transit delay (ISO/IEC 15802-3, 6.3.6) will not be exceeded at the time at which the frame would subsequently be transmitted.

A frame queued for transmission on a Port shall be removed from that queue if the associated Port leaves the forwarding state.

Removal of a frame from a queue for any particular Port does not of itself imply that it shall be removed from a queue for transmission on any other Port.

8.7.4 Selecting frames for transmission

The following algorithm shall be supported by all Bridges as the default algorithm for selecting frames for transmission:

a) For each Port, frames are selected for transmission on the basis of the traffic classes that the Port supports. For a given supported value of traffic class, frames are selected from the corresponding queue for transmission only if all queues corresponding to numerically higher values of traffic class supported by the Port are empty at the time of selection;
b) For a given queue, the order in which frames are selected for transmission shall maintain the ordering requirement specified in 8.7.3.

Additional algorithms, selectable by management means, may be supported as an implementation option so long as the requirements of 8.7.3 are met.

8.7.5 Mapping priority

The user_priority parameter in an EM_UNITDATA.request primitive (7.1) shall be equal to the user_priority parameter in the corresponding data indication.

The mapping of user_priority to outbound access_priority is achieved via fixed, MAC method-specific mappings. The access_priority parameter in an EM_UNITDATA.request primitive (7.1) shall be determined from the user_priority in accordance with the values shown in Table 8-3 for the MAC methods that will carry the data request. The values shown in Table 8-3 are not modifiable by management or other means.

The table shows two columns for the 8802-5 MAC method. The mapping in the column marked “8802-5 (alternate)” is included in order to permit backwards compatibility with equipment manufactured in accordance with ISO/IEC 10038: 1993; however, the use of this mapping reduces the number of available access priority values to three. For this reason, it is recommended that the column marked “8802-5 (default)” is supported as the default mapping where backward compatibility is not an issue.

8.7.6 Recalculating FCS

Where a frame is being forwarded between two individual MAC Entities of the same IEEE 802 LAN type, and relaying the frame involves no changes to the data that is within the FCS coverage, the FCS received in the EM_UNITDATA.indication primitive may be supplied in the corresponding EM_UNITDATA.request primitive and not recalculated (7.1, 7.2, ISO/IEC 15802-3, 6.3.7).

Where a frame is being forwarded between two individual MAC Entities of different types, recalculation of the FCS is necessary if the differences between the LAN MAC methods is such that an FCS calculated according to the MAC procedures for the destination MAC method would differ from the FCS carried by the

Table 8-3—Outbound access priorities

user_priority	Outbound Access Priority per MAC method								
	802.3	8802-4	8802-5 (default)	8802-5 (alternate)	8802-6	802.9a*	8802.11	8802-12	FDDI
0	0	0	0	4	0	0	0	0	0
1	0	1	1	4	1	0	0	0	1
2	0	2	2	4	2	0	0	0	2
3	0	3	3	4	3	0	0	0	3
4	0	4	4	4	4	0	0	4	4
5	0	5	5	5	5	0	0	4	5
6	0	6	6	6	6	0	0	4	6
7	0	7	6	6	7	0	0	4	6

*In the absence of a definition, in ISO/IEC 15802-3, 6.5, of support by IEEE Std 802.9a-1995, it is assumed that for this MAC method, access priority 0 will map to “low.”

received frame, or if relaying the frame involves changes to the data that is within the FCS coverage. Where necessary, the FCS is recalculated according to the specific MAC procedures of the transmitting MAC entity.

NOTE—There are two possibilities for recreating a valid FCS. The first is to generate a new FCS by algorithmically modifying the received FCS, based on knowledge of the FCS algorithm and the transformations that the frame has undergone between reception and transmission. The second is to rely on the normal MAC procedures to recalculate the FCS for the outgoing frame. The former approach may be preferable in terms of its ability to protect against increased levels of undetected frame errors. ISO/IEC 15802-3, Annex G, discusses these possibilities in more detail. The frame_check_sequence parameter of the Enhanced Internal Sublayer Service (7.1) is able to signal the validity, or otherwise, of the FCS; an unspecified value in this parameter in a data request indicates to the transmitting MAC that the received FCS is no longer valid, and the FCS must therefore be recalculated.

FCS recalculation is necessary if any of the following conditions are true:

a) The algorithm used to determine the FCS differs between the MAC methods used by the two MAC entities;
b) The FCS coverage differs between the MAC methods used by the two MAC entities;
c) Relaying the frame between the two MAC entities involves changes to the data that is within the coverage of the FCS (e.g., the frame was tagged on one link, but not on the other).

8.8 The egress rules

Frames shall be filtered, i.e., discarded, if

a) For the frame’s VID, as determined by the ingress rules (8.6), the transmission Port is not present in the Member set (8.11.9); or
b) The value of the include_tag parameter, determined as shown below, is False, and the Bridge does not support the ability to translate embedded MAC Address information from the format indicated

by the canonical_format_indicator parameter to the format appropriate to the MAC method on which the data request will be carried.

NOTE 1—The meanings of the terms Canonical format and Non-canonical format are discussed in Annex F.

The value of the include_tag parameter in the data request primitive is determined as follows:

c) If, for the frame's VID, as determined by the ingress rules (8.6), the transmission Port is present in the untagged set (8.11.9), then the value False is used. Otherwise;
d) The value True is used.

NOTE 2—As all incoming frames, including priority-tagged frames, are classified as belonging to a VLAN by the ingress rules (8.6), the transmitting Port only transmits VLAN-tagged frames or untagged frames, and can never transmit priority-tagged frames. Hence, a station sending a priority-tagged frame via a VLAN Bridge will receive a response that is either VLAN-tagged or untagged, depending upon the state of the untagged set for the VLAN concerned.

The value of the canonical_format_indicator parameter of the data request primitive is equal to the value of that parameter as received in the corresponding data indication.

8.9 Frame transmission

The individual MAC Entity associated with each Bridge Port transmits frames submitted to it by the MAC Relay Entity.

Relayed frames are submitted for transmission by the Forwarding Process. The EM_UNITDATA.request primitive associated with such frames conveys the values of the source and destination address fields received in the corresponding EM_UNITDATA.indication primitive.

LLC Protocol Data Units are submitted by LLC as a user of the MAC Service provided by the Bridge Port. Frames transmitted to convey such Protocol Data Units carry the individual MAC Address of the Port in the source address field.

Each frame is transmitted subject to the MAC procedures to be observed for that specific IEEE 802 LAN technology. The values of the frame_type and mac_action parameters of the corresponding EM_UNITDATA.request primitive shall be user_data_frame and request_with_no_response respectively (7.2; ISO/IEC 15802-3, 6.5).

Frames transmitted following a request by the LLC user of the MAC Service provided by the Bridge Port shall also be submitted to the MAC Relay Entity.

8.10 The Learning Process

The Learning Process observes the source MAC Addresses of frames received on each Port and updates the Filtering Database conditionally on the state of the receiving Port. The VID associated with the frame is used to ensure that the address information is learned relative to the frame's VLAN.

Frames are submitted to the Learning Process by the ingress rules as specified in 8.6.

The Learning Process can deduce the Port through which particular end stations in the Bridged LAN can be reached by inspection of the source MAC Address field and VID of received frames. It records such information in the Filtering Database (8.11). It shall create or update a Dynamic Filtering Entry (8.11.3) associated with the frame's VID (8.11.9), associating the reception Port with the source MAC Address, if and only if

a) The Port on which the frame was received is in a state that allows learning (ISO/IEC 15802-3, 8.4), and
b) The source address field of the frame denotes a specific end station, i.e., is not a group MAC Address, and
c) The resulting number of entries would not exceed the capacity of the Filtering Database, and
d) The Member set (8.11.9) for the frame's VID includes at least one Port.

NOTE—If the Member set for a given VID is the empty set, then that VLAN is not currently active, and the Bridge will therefore filter all frames destined for that VLAN, regardless of their destination address. There is therefore no reason to include MAC Address filtering information in the Filtering Database for that VLAN until such a time as it becomes active.

If the Filtering Database is already filled up to its capacity, but a new entry would otherwise be made, then an existing entry may be removed to make room for the new entry.

Figure 8-5 illustrates the operation of the Learning Process in the inclusion of station location information carried by a single frame, received on one of the Ports of a Bridge, in the Filtering Database.

8.11 The Filtering Database

The Filtering Database supports queries by the Forwarding Process as to whether frames received by the Forwarding Process, with given values of destination MAC Address parameter and VID, are to be forwarded through a given potential transmission Port (8.7.1 and 8.7.2). It contains filtering information in the form of filtering entries that are either

a) Static, and explicitly configured by management action; or
b) Dynamic, and automatically entered into the Filtering Database by the normal operation of the bridge and the protocols it supports.

Two entry types are used to represent static filtering information. The Static Filtering Entry represents static information in the Filtering Database for individual and for group MAC Addresses. It allows administrative control of

c) Forwarding of frames with particular destination addresses; and
d) The inclusion in the Filtering Database of dynamic filtering information associated with Extended Filtering Services, and use of this information.

The Filtering Database shall contain entries of the Static Filtering Entry type.

The Static VLAN Registration Entry represents all static information in the Filtering Database for VLANs. It allows administrative control of

e) Forwarding of frames with particular VIDs;
f) The inclusion/removal of tag headers in forwarded frames; and
g) The inclusion in the Filtering Database of dynamic VLAN membership information, and use of this information.

The Filtering Database may contain entries of the Static VLAN Registration Entry type.

Static filtering information is added to, modified, and removed from the Filtering Database only under explicit management control. It shall not be automatically removed by any ageing mechanism. Management of static filtering information may be carried out by use of the remote management capability provided by Bridge Management (8.13) using the operations specified in Clause 12.

Three entry types are used to represent dynamic filtering information. Dynamic Filtering Entries are used to specify the Ports on which individual MAC Addresses have been learned. They are created and updated by the Learning Process (8.10), and are subject to ageing and removal by the Filtering Database. Group Registration Entries support the registration of group MAC Addresses. They are created, updated, and removed by the GMRP protocol in support of Extended Filtering Services (8.11.4; ISO/IEC 15802-3, 6.6.5; ISO/IEC 15802-3, Clause 10). Dynamic VLAN Registration Entries are used to specify the Ports on which VLAN membership has been dynamically registered. They are created, updated, and removed by the GVRP protocol, in support of automatic VLAN membership configuration (Clause 11).

Static Filtering Entries and Group Registration Entries comprise

h) A MAC Address specification;
i) A VLAN Identifier (VID);
j) A Port Map, with a control element for each outbound Port to specify filtering for that MAC Address specification and VID.

Dynamic Filtering Entries comprise

k) A MAC Address specification;
l) A locally significant Filtering Identifier (FID; see 8.11.7);
m) A Port Map, with a control element for each outbound Port to specify filtering for that MAC Address specification in the VLAN(s) allocated to that FID.

Static and Dynamic VLAN Registration Entries comprise

n) A VLAN Identifier;
o) A Port Map, with a control element for each outbound Port to specify filtering for the VLAN.

Dynamic filtering information may be read by use of the remote management capability provided by Bridge Management (8.13) using the operations specified in Clause 12.

The Filtering Services supported by a Bridge (Basic and Extended Filtering Services) determine the default behavior of the Bridge with respect to the forwarding of frames destined for group MAC Addresses. In Bridges that support Extended Filtering Services, the default forwarding behavior for group MAC Addresses, for each Port, and for each VID, can be configured both statically and dynamically by means of Static Filtering Entries and/or Group Registration Entries that can carry the following MAC Address specifications:

p) All Group Addresses, for which no more specific Static Filtering Entry exists;
q) All Unregistered Group Addresses (i.e., all group MAC Addresses for which no Group Registration Entry exists), for which no more specific Static Filtering Entry exists.

NOTE 1—The All Group Addresses specification p) above, when used in a Static Filtering Entry with an appropriate control specification, provides the ability to configure a Bridge that supports Extended Filtering Services to behave as a Bridge that supports only Basic Filtering Services on some or all of its Ports. This might be done for the following reasons:

— The Ports concerned serve "legacy" devices that wish to receive multicast traffic, but are unable to register Group membership;
— The Ports concerned serve devices that need to receive all multicast traffic, such as routers or diagnostic devices.

The Filtering Database shall support the creation, updating and removal of Dynamic Filtering Entries by the Learning Process (8.10). In Bridges that support Extended Filtering Services, the Filtering Database shall support the creation, updating, and removal of Group Registration Entries by GMRP (ISO/IEC 15802-3, Clause 10).

Figure 8-4 illustrates use of the Filtering Database by the Forwarding Process in a single instance of frame relay between the Ports of a Bridge with two Ports.

Figure 8-5 illustrates the creation or update of a dynamic entry in the Filtering Database by the Learning Process. The entries in the Filtering Database allow MAC Address information to be learned independently for each VLAN or set of VLANs, by relating a MAC Address to the VLAN or set of VLANs on which that address was learned. This has the effect of creating independent Filtering Databases for each VLAN or set of VLANs that is present in the Bridged LAN.

NOTE 2—This standard specifies a single Filtering Database that contains all Filtering Database entries; however, the inclusion of VIDs and FIDs in the filtering entries effectively provides distinct ISO/IEC 15802-3-style Filtering Databases per VLAN or set of VLANs.

NOTE 3—The ability to create VLAN-dependent Filtering Database entries allows a VLAN Bridge to support

— Multiple end stations with the same individual MAC Address residing on different VLANs;
— End stations with multiple interfaces, each using the same individual MAC Address,
as long as not more than one end station or interface that uses a given MAC Address resides in a given VLAN.

Figure 8-6 illustrates the operation of the Bridge Protocol Entity (8.12), which operates the Spanning Tree Algorithm and Protocol, and its notification of the Filtering Database of changes in active topology signaled by that protocol.

There are no standardized constraints on the size of the Filtering Database in an implementation for which conformance to this standard is claimed. The PICS Proforma in Annex A requires the following to be specified for a given implementation:

r) The total number of entries (Static Filtering Entries, Dynamic Filtering Entries, Group Registration Entries, Static VLAN Registration Entries, and Dynamic VLAN Registration Entries) that the implementation of the Filtering Database can support, and
s) Of that total number, the total number of VLAN Registration Entries (static and dynamic) that the Filtering Database can support.

8.11.1 Static Filtering Entries

A Static Filtering Entry specifies

a) A MAC Address specification, comprising
   1) An Individual MAC Address; or
   2) A group MAC Address; or
   3) All Group Addresses, for which no more specific Static Filtering Entry exists; or
   4) All Unregistered Group Addresses, for which no more specific Static Filtering Entry exists.
b) The VID of the VLAN to which the static filtering information applies;
c) A Port Map, containing a control element for each outbound Port, specifying that a frame with a destination MAC Address and VID that meets this specification is to be
   1) Forwarded, independently of any dynamic filtering information held by the Filtering Database; or
   2) Filtered, independently of any dynamic filtering information; or
   3) Forwarded or filtered on the basis of dynamic filtering information, or on the basis of the default Group filtering behavior for the outbound Port (8.11.6) if no dynamic filtering information is present specifically for the MAC Address.

All Bridges shall have the capability to support the first two values for the MAC Address specification, and all three values for each control element for all Static Filtering Entries (i.e., shall have the capability to support a1, a2, c1, c2, and c3 above).

A Bridge that supports Extended Filtering Services shall have the capability to support all four values for the MAC Address specification and all three control element values for all Static Filtering Entries.

For a given MAC Address specification, a separate Static Filtering Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

In addition to controlling the forwarding of frames, Static Filtering Entries for group MAC Addresses provide the Registrar Administrative Control values for the GMRP protocol (ISO/IEC 15802-3, Clauses 10, 12, and 12.9.1). Static configuration of forwarding of specific group addressed frames to an outbound port indicates Registration Fixed on that port: a desire to receive frames addressed to that Group even in the absence of dynamic information. Static configuration of filtering of frames that might otherwise be sent to an outbound port indicates Registration Forbidden. The absence of a Static Filtering Entry for the group address, or the configuration of forwarding or filtering on the basis of dynamic filtering information, indicates Normal Registration.

8.11.2 Static VLAN Registration Entries

A Static VLAN Registration Entry specifies

a) The VID of the VLAN to which the static filtering information applies;
b) A Port Map, consisting of a control element for each outbound Port, specifying
   1) The Registrar Administrative Control values for the GVRP protocol (Clause 11) for the VLAN specified. In addition to providing control over the operation of GVRP, these values can also directly affect the forwarding behavior of the Bridge, as described in 8.11.9. The values that can be represented are
      i) Registration Fixed; or
      ii) Registration Forbidden; or
      iii) Normal Registration.
   2) Whether frames destined for the VLAN specified are to be VLAN-tagged or untagged when forwarded through this Port.

All Bridges shall be capable of supporting all values for each control element for all Static VLAN Registration Entries; however, the ability to support more than one untagged VLAN on egress on any given Port is optional (see 5.1 and 5.2).

NOTE—In other words, it shall be possible to configure any VLAN as untagged on egress, but it is an implementation option as to whether only a single untagged VLAN per Port on egress is supported, or whether multiple untagged VLANs per Port on egress are supported.

A separate Static VLAN Registration Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

8.11.3 Dynamic Filtering Entries

A Dynamic Filtering Entry specifies

a) An individual MAC Address;
b) The FID, an identifier assigned by the MAC Bridge (8.11.7) to identify a set of VIDs for which no more than one Dynamic Filtering Entry can exist for any individual MAC Address;

NOTE 1—An FID identifies a set of VLANs among which *Shared VLAN Learning* (3.9) takes place. Any pair of FIDs identifies two sets of VLANs between which *Independent VLAN Learning* (3.5) takes place. The allocation of FIDs by a Bridge is described in 8.11.7.

c) A Port Map that specifies forwarding of frames destined for that MAC Address and FID to a single Port.

NOTE 2—This is equivalent to specifying a single port number; hence, this specification is directly equivalent to the specification of dynamic entries in ISO/IEC 10038: 1993.

Dynamic Filtering Entries are created and updated by the Learning Process (8.10). They shall be automatically removed after a specified time, the Ageing Time, has elapsed since the entry was created or last updated. No more than one Dynamic Filtering Entry shall be created in the Filtering Database for a given combination of MAC Address and FID.

Dynamic Filtering Entries cannot be created or updated by management.

NOTE 3—Dynamic Filtering Entries may be read by management (Clause 12). The FID is represented in the management Read operation by any one of the VIDs that it represents. For a given VID, the set of VIDs that share the same FID may also be determined by management.

The ageing out of Dynamic Filtering Entries ensures that end stations that have been moved to a different part of the Bridged LAN will not be permanently prevented from receiving frames. It also takes account of changes in the active topology of the Bridged LAN that can cause end stations to appear to move from the point of view of the bridge; i.e., the path to those end stations subsequently lies through a different Bridge Port.

The Ageing Time may be set by management (Clause 12). A range of applicable values and a recommended default is specified in Table 8-4; this is suggested to remove the need for explicit configuration in most cases. If the value of Ageing Time can be set by management, the Bridge shall have the capability to use values in the range specified, with a granularity of 1 s.

Table 8-4—Ageing time parameter value

Parameter	Recommended default value	Range
Ageing time	300.0 s	10.0–1 000 000.0 s

NOTE 4—The granularity is specified in order to establish a common basis for the granularity expressed in the management operations defined in Clause 12, not to constrain the granularity of the actual timer supported by a conformant implementation. If the implementation supports a granularity other than 1 s, then it is possible that the value read back by management following a Set operation will not match the actual value expressed in the Set.

The Spanning Tree Algorithm and Protocol specified in ISO/IEC 15802-3, Clause 8, includes a procedure for notifying all Bridges in the Bridged LAN of topology change. It specifies a short value for the Ageing Timer, to be enforced for a period after any topology change (ISO/IEC 15802-3, 8.3.5). While the topology is not changing, this procedure allows normal ageing to accommodate extended periods during which addressed end stations do not generate frames themselves, perhaps through being powered down, without sacrificing the ability of the Bridged LAN to continue to provide service after automatic configuration.

8.11.4 Group Registration Entries

A Group Registration Entry specifies

a) A MAC Address specification, comprising
   1) A group MAC Address; or

   2) All Group Addresses, for which no more specific Static Filtering Entry exists; or
   3) All Unregistered Group Addresses, for which no more specific Static Filtering Entry exists.

b) The VID of the VLAN in which the dynamic filtering information was registered;
c) A Port Map, consisting of a control element for each outbound Port, which specifies forwarding (Registered) or filtering (Not registered) of frames destined to the MAC Address and VID.

Group Registration Entries are created, modified and deleted by the operation of GMRP (ISO/IEC 15802-3, Clause 10, as modified by Clause 10 of this standard). No more than one Group Registration Entry shall be created in the Filtering Database for a given combination of MAC Address specification and VID.

NOTE—It is possible to have a Static Filtering Entry which has values of Forward or Filter on some or all Ports that mask the dynamic values held in a corresponding Group Registration Entry. The values in the Group Registration Entry will continue to be updated by GMRP; hence, subsequent modification of that entry to allow the use of dynamic filtering information on one or more Ports immediately activates the true GMRP registration state that was hitherto masked by the static information.

8.11.5 Dynamic VLAN Registration Entries

A Dynamic VLAN Registration Entry specifies

a) The VID of the VLAN to which the dynamic filtering information applies;
b) A Port Map with a control element for each outbound Port specifying whether the VLAN is registered on that Port.

A separate Dynamic VLAN Registration Entry with a distinct Port Map may be created for each VLAN from which frames are received by the Forwarding Process.

8.11.6 Default Group filtering behavior

Forwarding and filtering of group-addressed frames may be managed by specifying defaults for each VLAN and outbound Port. The behavior of each of these defaults, as modified by the control elements of more explicit Filtering Database entries applicable to a given frame's MAC Address, VLAN classification, and outbound Port is as follows:

NOTE 1—As stated in 8.11.1, for a given MAC Address there may be separate Static Filtering Entries with a distinct Port Map for each VLAN.

a) *Forward All Groups.* The frame is forwarded, unless an explicit Static Filtering Entry specifies filtering independent of any dynamic filtering information.
b) *Forward Unregistered Groups.* The frame is forwarded, unless
   1) An explicit Static Filtering Entry specifies filtering independent of any dynamic filtering information; or
   2) An explicit Static Filtering Entry specifies forwarding or filtering on the basis of dynamic filtering information, and an applicable explicit Group Registration Entry exists specifying filtering; or
   3) An applicable explicit Static Filtering Entry does not exist, but an applicable Group Registration entry specifies filtering.
c) *Filter Unregistered Groups.* The frame is filtered unless
   1) An explicit Static Filtering Entry specifies forwarding independent of any dynamic filtering information; or
   2) An explicit Static Filtering Entry specifies forwarding or filtering on the basis of dynamic filtering information, and an applicable explicit Group Registration Entry exists specifying forwarding; or
   3) An applicable explicit Static Filtering Entry does not exist, but an applicable Group Registration entry specifies forwarding.

In Bridges that support only Basic Filtering Services, the default Group filtering behavior is Forward All Groups for all Ports of the Bridge, for all VLANs.

NOTE 2—Forward All Groups corresponds directly to the behavior specified in ISO/IEC 10038: 1993 when forwarding group MAC Addressed frames for which no static filtering information exists in the Filtering Database. Forward All Groups makes use of information contained in Static Filtering Entries for specific group MAC Addresses, but overrides any information contained in Group Registration Entries. Forward Unregistered Groups is analogous to the forwarding behavior of a Bridge with respect to individual MAC Addresses. If there is no static or dynamic information for a specific group MAC Address, then the frame is forwarded; otherwise, the frame is forwarded in accordance with the statically configured or dynamically learned information.

In Bridges that support Extended Filtering Services, the default Group filtering behavior for each outbound Port for each VLAN is determined by the following information contained in the Filtering Database:

d) Any Static Filtering Entries applicable to that VLAN with a MAC Address specification of All Group Addresses or All Unregistered Group Addresses;
e) Any Group Registration Entries applicable to that VLAN with a MAC Address specification of All Group Addresses or All Unregistered Group Addresses.

The means whereby this information determines the default Group filtering behavior is specified in 8.11.8, Table 8-6, and Table 8-7.

NOTE 3—The result is that the default Group filtering behavior for each VLAN can be configured for each Port of the Bridge via Static Filtering Entries, determined dynamically via Group Registration Entries created/updated by GMRP (Clause 10), or both. For example, in the absence of any static or dynamic information in the Filtering Database for All Group Addresses or All Unregistered Group Addresses, the default Group filtering behavior will be Filter Unregistered Groups on all Ports, for all VLANs. Subsequently, the creation of a Dynamic Group Registration Entry for All Unregistered Group Addresses indicating “Registered” for a given VLAN on a given Port would cause that Port to exhibit Forward Unregistered Groups behavior for that VLAN. Similarly, creating a Static Filtering Entry for All Group Addresses indicating “Registration Fixed” on a given Port for that VLAN would cause that Port to exhibit Forward All Groups behavior.

Hence, by using appropriate combinations of “Registration Fixed,” “Registration Forbidden,” and “Normal Registration” in the Port Maps of Static Filtering Entries for the All Group Addresses and All Unregistered Group Addresses address specifications, it is possible, for a given Port and VLAN, to

— Fix the default Group filtering behavior to be just one of the three behaviors described above; or
— Restrict the choice of behaviors to a subset of the three, and allow GMRP registrations (or their absence) to determine the final choice; or
— Allow any one of the three behaviors to be adopted, in accordance with any registrations received via GMRP.

8.11.7 Allocation of VIDs to FIDs

The allocation of VIDs to FIDs within a Bridge determines how learned individual MAC Address information is used in forwarding/filtering decisions within a Bridge; whether such learned information is confined to individual VLANs, shared among all VLANs, or confined to specific sets of VLANs.

The allocation of VIDs to FIDs is determined on the basis of

a) The set of *VLAN Learning Constraints* that have been configured into the Bridge (by means of the management operations defined in Clause 12);
b) Any fixed mappings of VIDs to FIDs that may have been configured into the Bridge (by means of the management operations defined in Clause 12);
c) The *set of active VLANs* (i.e., those VLANs on whose behalf the Bridge may be called upon to forward frames). A VLAN is active if either of the following is true:
    1) The VLAN’s Member set (8.11.9) contains one Port that is in a forwarding state, and at least one other Port of the Bridge is both in a forwarding state and has Ingress Filtering (8.4.5) disabled;

   2) The VLAN’s Member set contains two or more Ports that are in a forwarding state.

d) The capabilities of the Bridge with respect to the number of FIDs that it can support, and the number of VIDs that can be allocated to each FID.

A VLAN Bridge shall support a minimum of one FID, and may support up to 4094 FIDs. For the purposes of the management operations, FIDs are numbered from 1 through N, where N is the maximum number of FIDs supported by the implementation.

A VLAN Bridge shall support the ability to allocate at least one VID to each FID, and may support the ability to allocate more than one VID to each FID.

The number of VLAN Learning Constraints supported by a VLAN Bridge is an implementation option.

NOTE—In an SVL/IVL Bridge (3.11), a number of FIDs are supported, and one or more VID can be mapped to each FID. In an SVL Bridge (3.10), a single FID is supported, and all VIDs are mapped to that FID. In an IVL Bridge (3.6), a number of FIDs are supported, and only one VID can be mapped to each FID.

8.11.7.1 Fixed and dynamic VID to FID allocations

A Bridge may support the ability to define fixed allocations of specific VIDs to specific FIDs, via an allocation table that may be read and modified by means of the management operations defined in Clause 12. For each VID supported by the implementation, the allocation table indicates one of the following:

a) The VID is currently not allocated to any FID; or
b) A fixed allocation has been defined (via management), allocating this VID to FID X; or
c) A dynamic allocation has been defined (as a result of applying the VLAN Learning Constraints), allocating this VID to FID X.

For any VID that has no fixed allocation defined, the Bridge can dynamically allocate that VID to an appropriate FID, in accordance with the current set of VLAN Learning Constraints.

8.11.7.2 VLAN Learning Constraints

There are two types of VLAN Learning Constraint:

a) A Shared Learning Constraint (or S Constraint) asserts that Shared VLAN Learning shall occur between a pair of identified VLANs. S Constraints are of the form {A S B}, where A and B are VIDs. An S constraint is interpreted as meaning that Shared VLAN Learning shall occur between the VLANs identified by the pair of VIDs;
b) An Independent Learning Constraint (or I Constraint) asserts that a given VLAN is a member of a set of VLANs amongst which Independent VLAN Learning shall occur. I Constraints are of the form {A I N}, where A is a VID and N is an Independent Set Identifier. An I Constraint is interpreted as meaning that Independent VLAN Learning shall occur among the set of VLANs comprising VLAN A and all other VLANs identified in I Constraints that carry the same Independent Set Identifier, N.

A given VID may appear in any number (including zero) of S Constraints and/or I Constraints.

NOTE 1—S Constraints are

— *Symmetric*: e.g., {A S B} and {B S A} both express an identical constraint;
— *Transitive*: e.g., {A S B}, {B S C} implies the existence of a third constraint, {A S C};
— *Reflexive*: e.g., {A S A} is a valid S Constraint.

I Constraints are not

— *Symmetric*: e.g., {A I 1} and {1 I A} express different constraints;
— *Transitive*: e.g., ({A I 1}, {B I 1}, {B I 2}, {C I 2}) does not imply either {A I 2} or {C I 1}.

The allocation of VIDs to FIDs shall be such that, for all members of the set of active VLANs (8.11.7),

c) A given VID shall be allocated to exactly one FID;
d) If a given VID appears in an I Constraint, then it shall not be allocated to the same FID as any other VID that appears in an I Constraint with the same Independent Set Identifier;
e) If a given VID appears in an S Constraint (either explicit, or implied by the transitive nature of the specification), then it shall be allocated to the same FID as the other VID identified in the same S Constraint;
f) If a VID does not appear in any S or I Constraints, then the Bridge may allocate that VID to any FID of its choice.

NOTE 2—The intent is that the set of Learning Constraints is defined on a global basis; i.e., that all VLAN-aware Bridges are configured with the same set of constraints (although individual constraints may well be defined and distributed by different managers/administrators). Any Bridge therefore sees the complete picture in terms of the Learning Constraints that apply to all VLANs present in the Bridged LAN, regardless of whether they all apply to VLANs that are present in that particular Bridge. This standard provides the definition, in Clause 12, of managed objects and operations that model how individual constraints can be configured in a Bridge; however, the issue of how a distributed management system might ensure the consistent setting of constraints in all Bridges in a Bridged LAN is not addressed by this standard.

8.11.7.3 VLAN Learning Constraint inconsistencies and violations

The application of the rules specified in 8.11.7.2, coupled with any fixed allocations of VIDs to FIDs that may have been configured, can result in the Bridge detecting Learning Constraint inconsistencies and/or violations (i.e., can result in situations where there are inherent contradictions in the combined specification of the VLAN Learning Constraints and the fixed allocations, or the Bridge’s own limitations mean that it cannot meet the set of VLAN Learning Constraints that have been imposed upon it).

A Bridge detects a Learning Constraint inconsistency if

a) The VLAN Learning Constraints, coupled with any fixed VID to FID allocations, are such that, if any given pair of VLANs became members of the set of active VLANs (8.11.7), the result would be a simultaneous requirement for Independent VLAN Learning and for Shared VLAN Learning for those two VLANs. Such an inconsistency would require the Bridge to allocate that pair of VIDs both to the same FID and to different FIDs.

Learning Constraint inconsistencies are detected when a management operation (12.10.3) attempts to set a new Learning Constraint value, or to modify the fixed VID to FID allocations. If the new constraint or allocation that is the subject of the operation is inconsistent with those already configured in the Bridge, then the management operation shall not be performed and an error response shall be returned.

A Bridge detects a Learning Constraint violation if

b) The Bridge does not support the ability to map more than one VID to any given FID, and the VLAN Learning Constraints indicate that two or more members of the active set of VLANs require to be mapped to the same FID; or
c) The number of FIDs required in order to correctly configure the Bridge to meet the VLAN Learning Constraints and fixed VID to FID allocations for all members of the active set of VLANs exceeds the number of FIDs supported by the Bridge.

Learning Constraint violations are detected

d) When a VLAN that was hitherto not a member of the set of active VLANs (8.11.7) becomes active, either as a result of management action or as a result of the operation of GVRP, resulting in the Bridge no longer being able to support the defined set of constraints and/or fixed allocations for the set of active VLANs; or

e) When other management reconfiguration actions, such as defining a new Learning Constraint or fixed VID to FID allocation, results in the Bridge no longer being able to support the defined set of constraints and/or fixed allocations for the set of active VLANs.

On detection of a violation, the Bridge issues the Notify Learning Constraint Violation management notification (12.10.3.10), in order to alert any management stations to the existence of the violation. There is the potential for a single change in configuration to result in more than one VLAN whose constraints cannot be met; in such cases, multiple notifications are generated.

8.11.8 Querying the Filtering Database

If a frame is classified into a VLAN containing a given outbound Port in its member set (8.11.9), forwarding or filtering through that Port is determined by the control elements of filtering entries for the frame's destination MAC Address and for VLANs with the same VID or Filtering Identifier (FID, 8.11.3) as the frame's VLAN.

Each entry in the Filtering Database for a MAC Address comprises

a) A MAC Address specification;

b) A VID or, in the case of Dynamic Filtering Entries, an FID;

c) A Port Map, with a control element for each outbound Port.

For Dynamic Filtering Entries, the FID that corresponds to a given VID is determined as specified in 8.11.7.

For a given VID, a given individual MAC Address specification can be included in the Filtering Database in a Static Filtering Entry, a Dynamic Filtering Entry, both or neither. Table 8-5 combines Static Filtering Entry and Dynamic Filtering Entry information for an individual MAC Address to specify forwarding, or filtering, of a frame with that destination MAC Address and VID through an outbound Port.

NOTE 1—The use of FID in this table for Static Filtering Entries, and the text in parentheses in the headings, reflects the fact that, where more than one VID maps to a given FID, there may be more than one Static Filtering Entry that affects the forwarding decision for a given individual MAC Address. The effect of all Static Filtering Entries for that address, and for VIDs that correspond to that FID, is combined, such that, for a given outbound Port:
— IF <any static entry for any VIDs that map to that FID specifies Forwarding> THEN <result = Forwarding>
— ELSE IF <any static entry for any VIDs that map to that FID specifies Filtering> THEN <result = Filtering>
— ELSE <result = Use Dynamic Filtering Information>

Table 8-6 specifies the result, Registered or Not Registered, of combining a Static Filtering Entry and a Group Registration Entry for the "All Group Addresses" address specification, and for the "All Unregistered Group Addresses" address specification for an outbound Port.

Table 8-7 combines Static Filtering Entry and Group Registration Entry information for a specific group MAC Address with the Table 8-6 results for All Group Addresses and All Unregistered Group Addresses to specify forwarding, or filtering, of a frame with that destination group MAC Address through an outbound Port.

Table 8-5—Combining Static and Dynamic Filtering Entries for an individual MAC Address

Filtering Information	Control Elements in any Static Filtering Entry or Entries for this individual MAC Address, FID, and outbound Port specify:				
	Forward (Any Static Filtering Entry for this Address/ FID/Port specifies Forward)	**Filter (No Static Filtering Entry for this Address/ FID/Port specifies Forward)**	**Use Dynamic Filtering Information (No Static Filtering Entry for this Address/FID/Port specifies Forward or Filter), or no Static Filtering Entry present. Dynamic Filtering Entry Control Element for this individual MAC Address, FID and outbound Port specifies:**		
			Forward	**Filter**	**No Dynamic Filtering Entry present**
Result	Forward	Filter	Forward	Filter	Forward

Table 8-6—Combining Static Filtering Entry and Group Registration Entry for "All Group Addresses" and "All Unregistered Group Addresses"

Filtering Information	Static Filtering Entry Control Element for this group MAC Address, VID, and outbound Port specifies:				
	Registration Fixed (Forward)	**Registration Forbidden (Filter)**	**Use Group Registration Information, or no Static Filtering Entry present. Group Registration Entry Control Element for this group MAC Address, VID and outbound Port specifies:**		
			Registered (Forward)	**Not Registered (Filter)**	**No Group Registration Entry present**
Result	Registered	Not Registered	Registered	Not Registered	Not Registered

Where a given VID is allocated to the same FID as one or more other VIDs, it is an implementation option as to whether

d) The results shown in Table 8-7 directly determine the forwarding/filtering decision for a given VID and group MAC Address (i.e., the operation of the Bridge with respect to group MAC Addresses ignores the allocation of VIDs to FIDs); or
e) The results for a given MAC Address and VID are combined with the corresponding results for that MAC Address for each other VID that is allocated to the same FID, so that if the Table 8-7 result is Forward in any one VLAN that shares that FID, then frames for that group MAC Address will be forwarded for all VLANs that share that FID (i.e., the operation of the Bridge with respect to group MAC Addresses takes account of the allocation of VIDs to FIDs).

NOTE 2—In case d), the implementation effectively operates a single FDB per VLAN for group MAC Addresses. In case e), the implementation combines static and registered information for group MAC Addresses in accordance with the VID to FID allocations currently in force, in much the same manner as for individual MAC Addresses.

Table 8-7—Forwarding or Filtering for specific group MAC Addresses

				Static Filtering Entry Control Element for this group MAC Address, VID and outbound Port specifies:				
				Registration Fixed (Forward)	Registration Forbidden (Filter)	Use Group Registration Information, or no Static Filtering Entry present. Group Registration Entry Control Element for this group MAC Address, VID and outbound Port specifies:		
						Registered (Forward)	Not Registered (Filter)	No Group Registration Entry present
All Group Addresses control elements for this VID and Port specify (Table 8-6):	Not Registered	All Unregistered Group Addresses control elements for this VID and Port specify (Table 8-6):	Not Registered	Forward	Filter	Forward	Filter	Filter (Filter Unregistered Groups)
			Registered	Forward	Filter	Forward	Filter	Forward (Forward Unregistered Groups)
	Registered			Forward	Filter	Forward (Forward All Groups)	Forward (Forward All Groups)	Forward (Forward All Groups)

8.11.9 Determination of the member set and untagged set for a VLAN

The VLAN configuration information contained in the Filtering Database for a given VLAN may include a Static VLAN Registration Entry (8.11.2) and/or a Dynamic VLAN Registration Entry (8.11.5). This information defines, for that VLAN:

a) The *member set*, consisting of the set of Ports through which members of the VLAN can currently be reached;

b) The *untagged set*, consisting of the set of Ports through which, if frames destined for the VLAN are to be transmitted, they shall be transmitted without tag headers. For all other Ports (i.e., all Ports that are not members of the untagged set), if frames destined for the VLAN are to be transmitted, they shall be transmitted with tag headers.

NOTE 1—As the operation of the ingress rules (8.6) always associates a non-null VLAN ID with an incoming frame, all frames (including received frames that were priority-tagged and carried the null VLAN ID in their tag header) will be transmitted with or without a tag header in accordance with the membership of the untagged set for their VID.

For a given VID, the Filtering Database can contain a Static VLAN Registration Entry, a Dynamic VLAN Registration Entry, both or neither. Table 8-8 combines Static VLAN Registration Entry and Dynamic VLAN Registration Entry information for a VLAN and Port to give a result *member*, or *not member*, for the Port. The member set for a given VLAN consists of all Ports for which the result is member.

Table 8-8—Determination of whether a Port is in a VLAN's member set

Filtering Information	Static VLAN Registration Entry Control Element for this VID and Port specifies:				
	Registration Fixed	Registration Forbidden	Normal Registration, or no Static VLAN Registration Entry present. Dynamic VLAN Registration Entry Control Element for this VID and Port specifies:		
			Registered	Not Registered	No Dynamic VLAN Registration Entry present
Result	Member	Not member	Member	Not member	Not member

Membership of the untagged set for a given VLAN is derived from Static VLAN Registration Entry information contained in the Filtering Database as follows:

c) If there is no Static VLAN Registration Entry for the VLAN, then the untagged set is the empty set; otherwise,

d) The untagged set is equal to the set of Ports for which the Port Map in the Static VLAN Registration Entry indicates that frames are to be transmitted untagged.

The untagged set and the member set for a given VLAN are used in determining the operation of the ingress rules (8.6) and the egress rules (8.8) for that VLAN.

The initial state of the Permanent Database contains a Static VLAN Registration Entry for the VLAN corresponding to the Default PVID (Table 9-2). The Port Map in this entry specifies Registration Fixed and forwarding untagged for all Ports of the Bridge. This entry may be modified or removed from the Permanent Database by means of the management operations defined in Clause 12 if the implementation supports these operations.

NOTE 2—This causes the default tagging state for the PVID to be untagged, and for all other VIDs to be tagged, unless otherwise configured; however, the management configuration mechanisms allow any VID (including the PVID) to be specified as VLAN-tagged or untagged on any Port. Under normal circumstances, the appropriate configuration for the PVID would be untagged on an access Port or a hybrid Port, and VLAN-tagged on a trunk Port (Annex D discusses the terms *access Port*, *hybrid Port*, and *trunk Port*).

8.11.10 Permanent Database

The Permanent Database provides fixed storage for a number of Static Filtering Entries and Static VLAN Registration Entries. The Filtering Database shall be initialized with the Filtering Database Entries contained in this fixed data store.

Entries may be added to and removed from the Permanent Database under explicit management control, using the management functionality defined in Clause 12. Changes to the contents of Static Filtering Entries or Static VLAN Registration Entries in the Permanent Database do not affect forwarding and filtering decisions taken by the Forwarding Process or the egress rules until such a time as the Filtering Database is re-initialized.

NOTE—This aspect of the Permanent Database can be viewed as providing a "boot image" for the Filtering Database, defining the contents of all initial entries, before any dynamic filtering information is added.

8.12 Bridge Protocol Entity and GARP Protocol Entities

The Bridge Protocol Entity operates the Spanning Tree Algorithm and Protocol.

The Bridge Protocol Entities of Bridges attached to a given individual LAN in a Bridged LAN communicate by exchanging Bridge Protocol Data Units (BPDUs).

Figure 8-6 illustrates the operation of the Bridge Protocol Entity including the reception and transmission of frames containing BPDUs, the modification of the state information associated with individual Bridge Ports, and notification of the Filtering Database of changes in active topology.

The GARP Protocol Entities operate the Algorithms and Protocols associated with the GARP Applications supported by the Bridge, and consist of the set of GARP Participants for those GARP Applications (ISO/IEC 15802-3, Clauses 10 and 12.3).

The GARP Protocol Entities of Bridges attached to a given individual LAN in a Bridged LAN communicate by exchanging GARP Protocol Data Units (GARP PDUs).

Figure 8-7 illustrates the operation of a GARP Protocol Entity including the reception and transmission of frames containing GARP PDUs, the use of control information contained in the Filtering Database, and notification of the Filtering Database of changes in filtering information.

8.13 Bridge Management

Remote management facilities may be provided by the Bridge. Bridge Management is modeled as being performed by means of the Bridge Management Entity. The facilities provided by Bridge Management, and the operations that support these facilities, are specified in Clause 12.

Bridge Management protocols use the MAC Service provided by the Bridged LAN.

8.14 Addressing

All MAC Entities communicating across a Bridged LAN shall use 48-bit addresses. These may be Universally Administered Addresses, Locally Administered Addresses, or a combination of both.

8.14.1 End stations

Frames transmitted between end stations using the MAC Service provided by a Bridged LAN carry the MAC Address of the source and destination end stations in the source and destination address fields of the frames, respectively. The address, or other means of identification, of a Bridge is not carried in frames transmitted between end stations for the purpose of frame relay in the Bridged LAN.

The broadcast address and other group MAC Addresses apply to the use of the MAC Service provided by a Bridged LAN as a whole. In the absence of explicit filters configured via management as Static Filtering Entries, or via GMRP as Group Registration Entries (8.11, Clause 12, and ISO/IEC 15802-3, Clause 10), frames with such destination addresses are relayed throughout the Bridged LAN.

8.14.2 Bridge Ports

The individual MAC Entity associated with each Bridge Port shall have a separate individual MAC Address. This address is used for any MAC procedures required by the particular MAC method employed.

Frames that are received from the LAN to which a Port is attached and that carry a MAC Address for the Port in the destination address field are submitted to the MAC Service User (LLC) exactly as for an end station.

8.14.3 Bridge Protocol Entities and GARP Protocol Entities

Bridge Protocol Entities only receive and transmit BPDUs. These are only received and transmitted from other Bridge Protocol Entities (or where two Bridge Ports are connected to the same LAN, to and from themselves).

GARP Protocol Entities only receive and transmit GARP PDUs (ISO/IEC 15802-3, 12.11) that are formatted according to the requirements of the GARP Applications they support. These are only received and transmitted from other GARP Protocol Entities.

A Bridge Protocol Entity or a GARP Protocol Entity uses the DL_UNITDATA.request primitive (see ISO/IEC 8802-2) provided by the individual LLC Entities associated with each active Bridge Port to transmit BPDUs or GARP PDUs. Each PDU is transmitted on one selected Bridge Port. PDUs are received through corresponding DL_UNITDATA.indication primitives. The source_address and destination_address parameters of the DL_UNITDATA.request primitive shall both denote the standard LLC address assigned to the Bridge Spanning Tree Protocol. This identifies the Bridge Protocol Entity and the GARP Protocol Entity among other users of LLC.

Each DL_UNITDATA.request primitive gives rise to the transmission of an LLC UI command PDU, which conveys the BPDU or GARP PDU in its information field. The source and destination LLC address fields are set to the values supplied in the request primitive.

The value assigned to the Bridge Spanning Tree Protocol LLC address is given in Table 8-9.[8]

Table 8-9—Standard LLC address assignment

Assignment	Value
Bridge spanning tree protocol	01000010

Code Representation: The least significant bit of the value shown is the right-most. The bits increase in significance from right to left. It should be noted that the code representation used here has been chosen in order to maintain consistency with the representation used elsewhere in this standard; however, it differs from the representation used in ISO/IEC 11802-1: 1997.

ISO/IEC 15802-3 defines a Protocol Identifier field, present in all BPDUs (ISO/IEC 15802-3, Clause 9) and GARP PDUs (ISO/IEC 15802-3, 12.11), which serves to identify different protocols supported by Bridge Protocol Entities and GARP Protocol Entities, within the scope of the LLC address assignment. This standard specifies a single value of the Protocol Identifier, defined in ISO/IEC 15802-3, Clause 9, for use in BPDUs. This value serves to identify BPDUs exchanged between Bridge Protocol Entities operating the Spanning Tree Algorithm and Protocol specified in ISO/IEC 15802-3, Clause 8. A second value of this protocol identifier for use in GARP PDUs is defined in ISO/IEC 15802-3, 12.11. This value serves to identify GARP PDUs exchanged between GARP Participants operating the GARP protocol specified in ISO/IEC 15802-3, Clause 12. Further values of this field are reserved for future standardization.

[8]ISO/IEC TR 11802-1: 1997, Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Technical reports and guidelines—Part 1: The structure and coding of Logical Link Control addresses in Local Area Networks, contains the full list of standard LLC address assignments, and documents the criteria for assignment.

A Bridge Protocol Entity or GARP Protocol Entity that receives a BPDU or a GARP PDU with an unknown Protocol Identifier shall discard that PDU.

A Bridge Protocol Entity that operates the Spanning Tree Algorithm and Protocol specified in ISO/IEC 15802-3, Clause 8, always transmits BPDUs addressed to all other Bridge Protocol Entities attached to the LAN on which the frame containing the BPDU is transmitted. A group address shall be used in the destination address field to address this group of Entities. This group address shall be configured in the Permanent Database (8.14.6) in order to confine BPDUs to the individual LAN on which they are transmitted.

A 48-bit Universal Address, known as the Bridge Group Address, has been assigned for this purpose. Its value is specified in Table 8-10. Bridges that use 48-bit Universally Administered Addresses shall use this address in the destination address field of all MAC frames conveying BPDUs.

Table 8-10—Reserved addresses

Assignment	Value
Bridge Group Address	01-80-C2-00-00-00
IEEE Std 802.3, 1998 Edition, Full Duplex PAUSE operation	01-80-C2-00-00-01
Reserved for future standardization	01-80-C2-00-00-02
Reserved for future standardization	01-80-C2-00-00-03
Reserved for future standardization	01-80-C2-00-00-04
Reserved for future standardization	01-80-C2-00-00-05
Reserved for future standardization	01-80-C2-00-00-06
Reserved for future standardization	01-80-C2-00-00-07
Reserved for future standardization	01-80-C2-00-00-08
Reserved for future standardization	01-80-C2-00-00-09
Reserved for future standardization	01-80-C2-00-00-0A
Reserved for future standardization	01-80-C2-00-00-0B
Reserved for future standardization	01-80-C2-00-00-0C
Reserved for future standardization	01-80-C2-00-00-0D
Reserved for future standardization	01-80-C2-00-00-0E
Reserved for future standardization	01-80-C2-00-00-0F

A GARP Protocol Entity that

a) Operates the GARP protocol specified in ISO/IEC 15802-3, Clause 12; and
b) Supports a given GARP Application,

always transmits GARP PDUs addressed to all other GARP Protocol Entities that

c) Implement the same GARP Application; and
d) Are attached to the LAN on which the frame containing the GARP PDU is transmitted.

A group MAC Address, specific to the GARP Application concerned, shall be used as the destination MAC Address field to address this group of GARP Protocol Entities. A set of 48-bit Universal Addresses, known as GARP Application addresses, have been assigned for that purpose. The values of the GARP Application addresses are defined in ISO/IEC 15802-3, Table 12-1. These group MAC Addresses are reserved for assignment to standard protocols, according to the criteria for such assignments (Clause 5.5 of ISO/IEC TR 11802-2).

NOTE—Table 11-1 allocates a group MAC Address for use by the GVRP application; however, the value allocated in that table is one of the GARP Application addresses reserved by ISO/IEC 15802-3, Table 12-1.

In Bridges that do not support any GARP applications, the set of GARP Application addresses should not be configured in the Filtering Database (8.11) or the Permanent Database (8.11.10). In Bridges that support one or more GARP applications, the set of GARP Application addresses should be configured as Static Filtering Entries in the Filtering Database (8.11.1) and Permanent Database (8.11.10) as follows:

e) GARP Application addresses assigned to GARP Applications that are supported by the Bridge should be configured in order to confine GARP PDUs for that GARP Application to the individual LAN on which they are transmitted;

f) GARP Application addresses assigned to GARP Applications that are not supported by the Bridge should not be configured in the Filtering Database or Permanent Database.

The source address field of MAC frames conveying BPDUs or GARP PDUs contains the individual MAC Address for the Bridge Port through which the PDU is transmitted (8.14.2).

8.14.4 Bridge Management Entities

Bridge Management Entities transmit and receive protocol data units using the Service provided by the individual LLC Entities associated with each Bridge Port. Each of these in turn uses the MAC Service, which is provided by the individual MAC Entities associated with that Port and supported by the Bridged LAN as a whole.

As a user of the MAC Service provided by a Bridged LAN, the Bridge Management Entity may be attached to any point in the Bridged LAN. Frames addressed to the Bridge Management Entity will be relayed by Bridges if necessary to reach the LAN to which it is attached.

In order to ensure that received frames are not duplicated, the basic requirement in a single LAN or a Bridged LAN that a unique address be associated with each point of attachment shall be met.

A Bridge Management Entity for a specific Bridge is addressed by one or more individual MAC Addresses in conjunction with the higher layer protocol identifier and addressing information. It may share one or more points of attachment to the Bridged LAN with the Ports of the Bridge with which it is associated. It is recommended that it make use of the MAC Service provided by all the MAC Entities associated with each Bridge Port, i.e., that it be reachable through each Bridge Port using frames carrying the individual MAC Address of that Port in the destination address field.

This standard specifies a standard group MAC Address for public use which serves to convey management requests to the Bridge Management Entities associated with all Bridge Ports attached to a Bridged LAN. A management request that is conveyed in a MAC frame carrying this address value in the destination address field will generally elicit multiple responses from a single Bridge. This address is known as the All LANs Bridge Management Group Address and takes the value specified in Table 8-11.

Table 8-11—Addressing bridge management

Assignment	Value
All LANs Bridge Management Group Address	01-80-C2-00-00-10

8.14.5 Unique identification of a Bridge

A unique 48-bit Universally Administered MAC Address, termed the Bridge Address, shall be assigned to each Bridge. The Bridge Address may be the individual MAC Address of a Bridge Port, in which case use of the address of the lowest numbered Bridge Port (Port 1) is recommended.

NOTE—The Spanning Tree Protocol (ISO/IEC 15802-3, Clause 8) requires that a single unique identifier be associated with each Bridge. That identifier is derived from the Bridge Address as specified in ISO/IEC 15802-3, 8.5.1.3, 8.5.3.7, and 9.2.5.

8.14.6 Reserved addresses

Frames containing any of the group MAC Addresses specified in Table 8-10 in their destination address field shall not be relayed by the Bridge. They shall be configured in the Permanent Database. Management shall not provide the capability to modify or remove these entries from the Permanent or the Filtering Databases. These group MAC Addresses are reserved for assignment to standard protocols, according to the criteria for such assignments (Clause 5.5 of ISO/IEC TR 11802-2).

8.14.7 Points of attachment and connectivity for Higher Layer Entities

Higher Layer Entities such as the Bridge Protocol Entity and GARP Protocol Entity (8.12), and Bridge Management (8.13) are modeled as being connected directly to the Bridged LAN via one or more points of attachment. From the point of view of their attachment to the Bridged LAN, Higher Layer Entities associated with a Bridge can be regarded as if they are distinct end stations, directly connected to one or more of the LAN segments served by the Bridge Ports, in the same way as any other end station is connected to the Bridged LAN. In practice, the Higher Layer Entities will, in many cases, share the same physical points of attachment used by the relay function of the Bridge, as stated in 8.14; however, from the point of view of the transmission and reception of frames by these functions, the behavior is the same as if they were contained in logically separate end stations with points of attachment “outside” the Port(s) with which they are associated. Figure 8-9 is functionally equivalent to Figure 8-3, but illustrates this logical separation between the points of attachment used by the Higher Layer Entities and points of attachment used by the MAC Relay Entity.

Higher Layer Entities fall into two distinct categories:

a) Those entities, such as the Bridge Management Entity, that require only a single point of attachment to the Bridged LAN;
b) Those entities, such as Bridge Protocol Entities and GARP Participants, that require a point of attachment per Port of the Bridge.

The fundamental distinction between these two categories is that for the latter, it is essential for the operation of the entity concerned that it is able to associate received frames with the LAN segment on which those frames were originally seen by the Bridge, and that it is able to transmit frames to peer entities that are connected directly to that LAN segment. It is therefore essential that

c) It does not receive frames via a point of attachment associated with one Port that have been relayed by the Bridge from other Ports; and
d) Frames that it transmits via one point of attachment are not relayed by the Bridge to any other Ports.

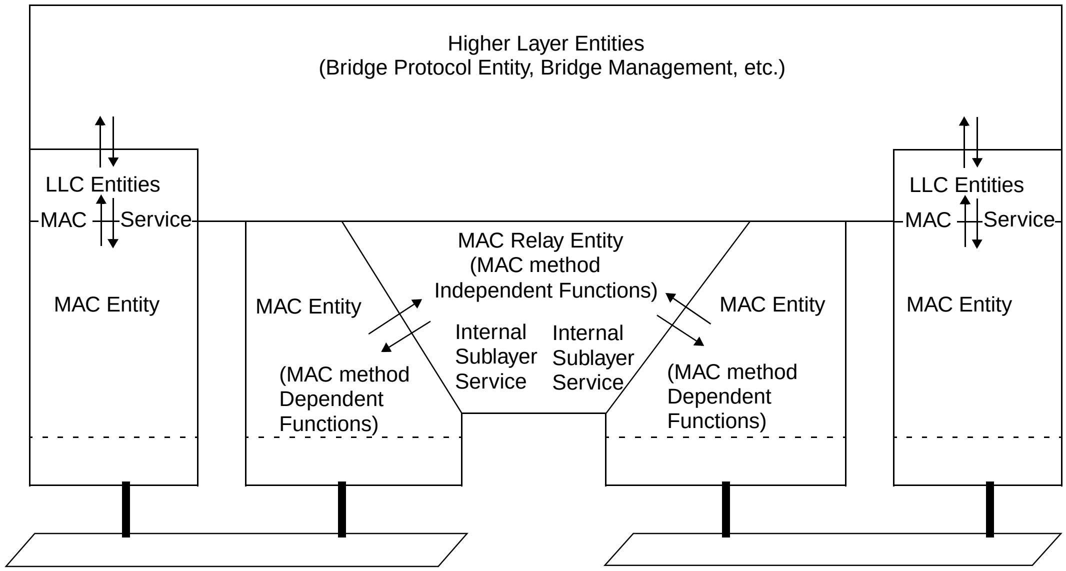


Figure 8-9—Logical separation of points of attachment used by Higher Layer Entities and the MAC Relay Entity

For this reason, the MAC Addresses used to reach entities of this type are permanently configured in the Filtering Database in order to prevent the Bridge from relaying such frames received via any Port to any other Port of the Bridge, as defined in 8.14.3 and 8.14.6.

NOTE 1—The MAC Addresses used to address such entities are generally group MAC Addresses.

The MAC Relay Entity forwards a frame received on one Port through the other Port(s) of the Bridge, subject to the following control information permitting such forwarding to take place:

e) The Port state information (8.4) associated with the Port on which the frame was received;
f) The information held in the Filtering Database (8.11);
g) The Port state information (8.4) associated with the Port(s) on which the frame is potentially to be transmitted.

This is illustrated in Figure 8-10, where the control information represented by the Port state and Filtering Database information is represented as a series of switches (shown in the open, disconnected state) inserted in the forwarding path provided by the MAC Relay Entity. For the Bridge to forward a given frame between two Ports, all three switches must be in the closed state. This figure also illustrates that the controls placed in the forwarding path have no effect upon the ability of a Higher Layer Entity to transmit and receive frames directly onto a given LAN segment via the point of attachment to that segment (e.g., from entity A to segment A); they only affect the path taken by any indirect transmission/reception (e.g., from entity A to segment B).

Figure 8-11 illustrates the state of the forwarding path with respect to frames destined for Higher Layer Entities that require per-Port points of attachment. The fact that the Filtering Databases in all Bridges are permanently configured to prevent relay of frames addressed to these entities means that they can receive frames only via their direct points of attachment (i.e., from segment A to entity A, and from segment B to entity B), regardless of the Port states.

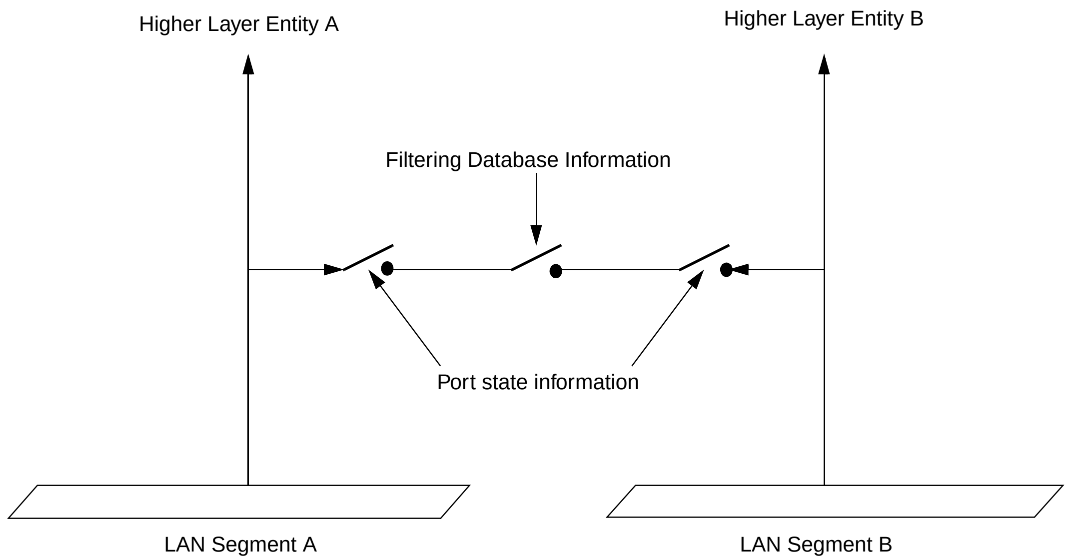


Figure 8-10—Effect of control information on the forwarding path

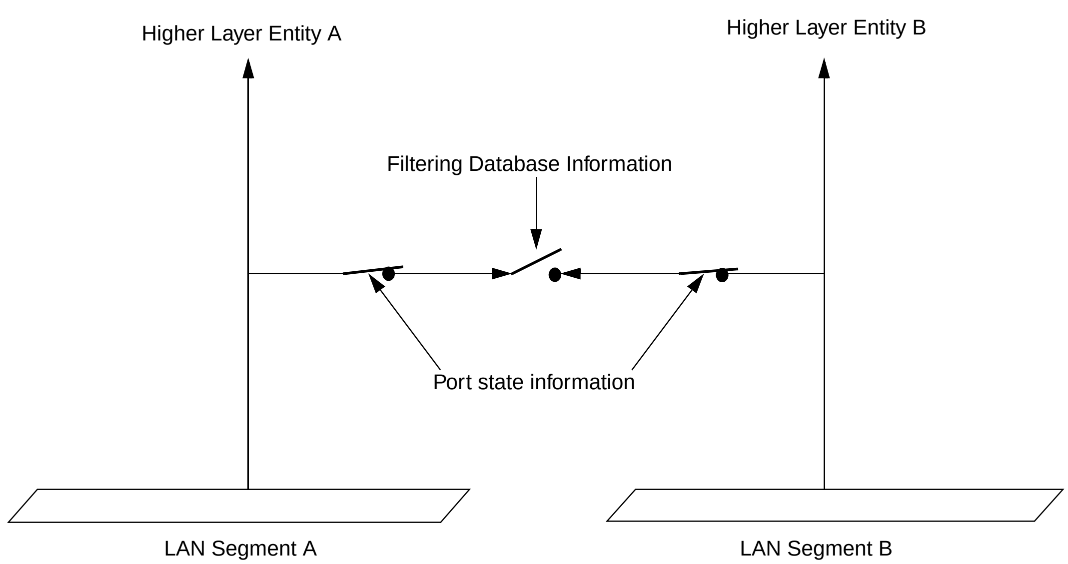


Figure 8-11—Per-Port points of attachment

Figure 8-12 illustrates the state of the forwarding path with respect to frames destined for a Higher Layer Entity that requires only a single point of attachment, for the case where the Port states and Filtering Database states permit relay of frames. Frames destined for the Higher Layer Entity that originate on LAN segment B are relayed by the Bridge, and are both received by the entity and transmitted on LAN segment A.

Figure 8-13 illustrates the state of the forwarding path with respect to frames destined for a Higher Layer Entity that requires only a single point of attachment, for the case where one of the Port states does not permit relay. Frames destined for the Higher Layer Entity that originate on LAN segment A are received by the entity; however, frames that originate on LAN segment B are not relayed by the Bridge, and can therefore only be received by the entity if there is some other forwarding path provided by other components of the Bridged LAN between segments A and B.

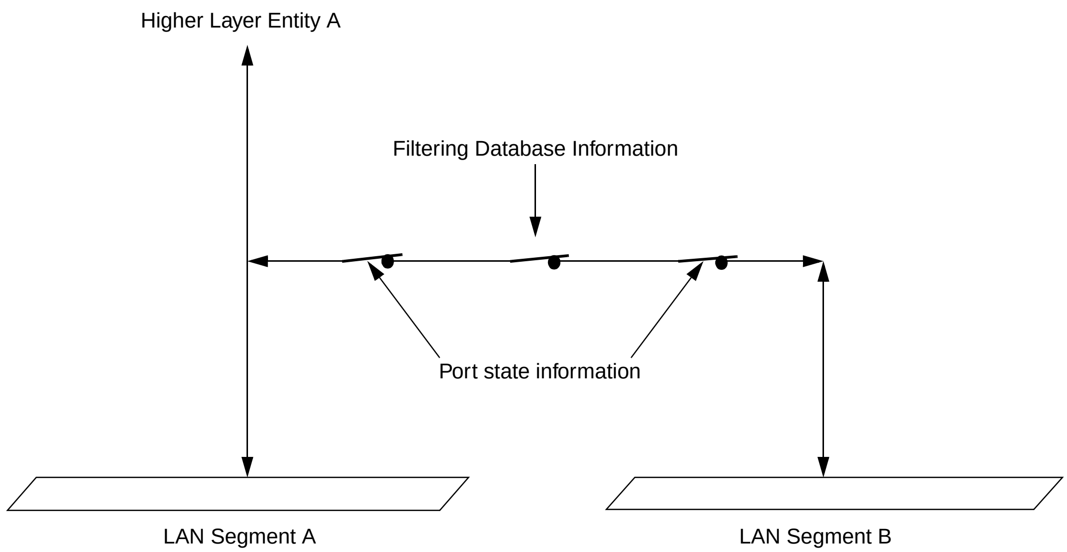


Figure 8-12—Single point of attachment—relay permitted

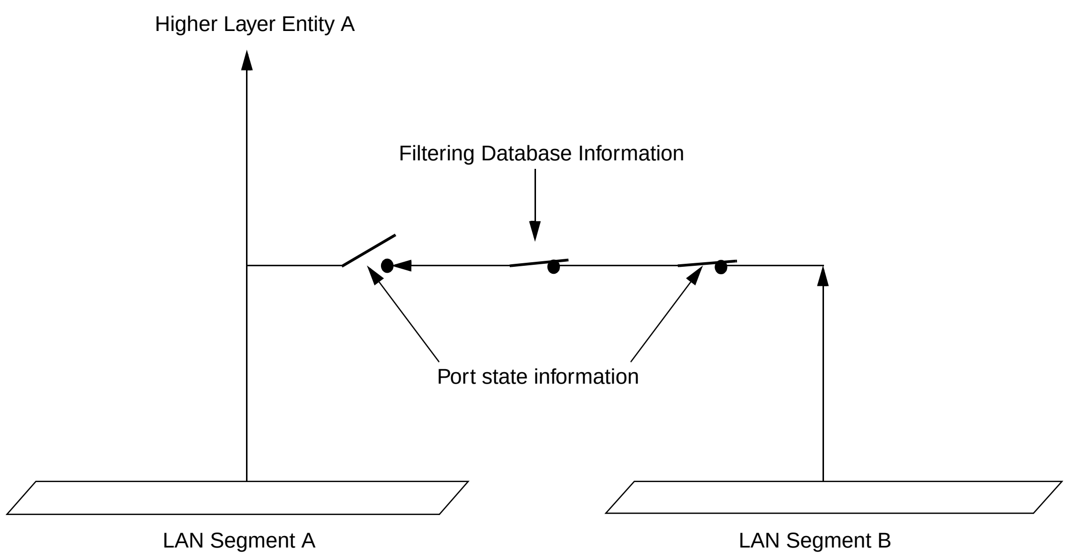


Figure 8-13—Single point of attachment—relay not permitted

NOTE 2—If the Port state shown in Figure 8-13 occurs as a result of the normal operation of the Spanning Tree (as opposed to being a result of equipment failure, or administrative control of Port state information), then such a path will exist, either via another Port of this Bridge (not shown in the diagram) connected to segment A, or via one or more Bridges providing a path between segments A and B. If there is no active Spanning Tree path from segment B to segment A, then the Bridged LAN has partitioned into two separate Bridged LANs, one on either side of this Port, and the Higher Layer Entity shown is only reachable via segment A.

In VLAN-aware Bridges, two more switches appear in the forwarding path, corresponding to the ingress and egress rules defined in 8.6 and 8.8, as illustrated in Figure 8-14.

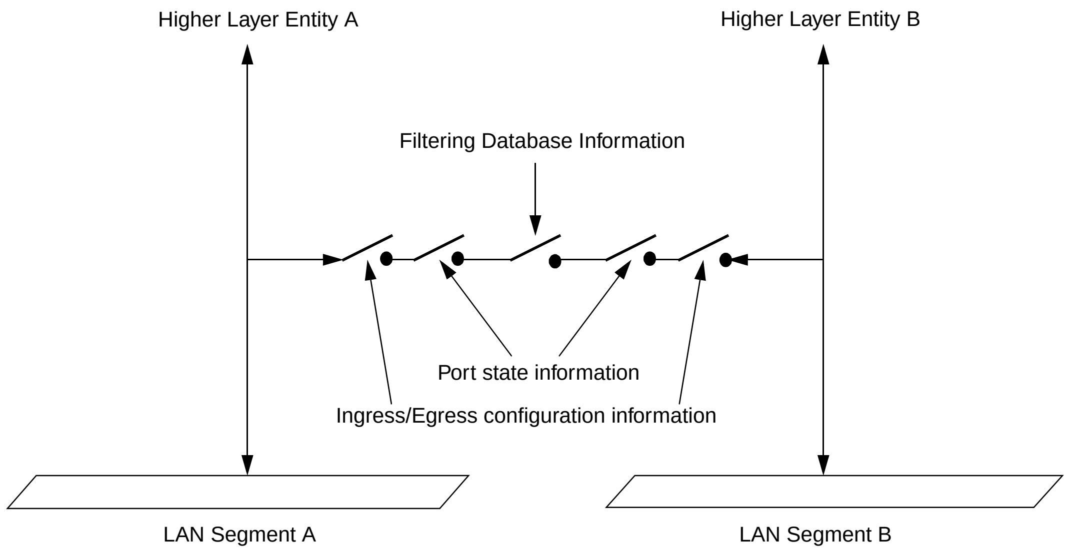


Figure 8-14—Ingress/egress control information in the forwarding path

As with Port state information, the configuration of the ingress and egress rules does not affect the reception of frames received on the same LAN segment as a Higher Layer Entity's point of attachment. For example, the reception of a frame by Higher Layer Entity A that was transmitted on LAN Segment A is unaffected by the ingress or egress configuration of either Port. However, for Higher Layer Entities that require only a single point of attachment, the ingress and egress configuration affects the forwarding path. For example, frames destined for Higher Layer Entity A that are transmitted on LAN Segment B would be subjected to the ingress rules that apply to Port B and the egress rules that apply to Port A.

The decision as to whether frames transmitted by Higher Layer Entities are VLAN-tagged or untagged depends upon the Higher Layer Entity concerned, and the connectivity that it requires

h) Spanning Tree BPDUs transmitted by the Bridge Protocol Entity are not forwarded by Bridges, and must be visible to all other BPEs attached to the same LAN segment. Such frames shall be transmitted untagged;

NOTE 3—Any BPDUs or GVRP PDUs that carry a tag header are not recognized as well-formed BPDUs or GVRP PDUs and are not forwarded by the Bridge.

i) The definition of the GVRP application (11.2.3) calls for all GVRP frames to be transmitted untagged for similar reasons;

j) The definition of the GMRP application (Clause 10) calls for all GMRP frames originating from VLAN-aware devices to be transmitted VLAN-tagged, in order for the VID in the tag to be used to identify the VLAN context in which the registration applies;

k) It may be necessary for PDUs transmitted for Bridge Management (8.13) to be VLAN-tagged in order to achieve the necessary connectivity for management in a VLAN Bridged LAN. In order to access a Bridge Management entity located in a region of the network that is served only by a given set of VLANs, it may be necessary to communicate with that entity using frames VLAN-tagged with one of the VIDs concerned, unless one of those VIDs also happens to be the PVID for the Port serving the management station.

9. Tagged frame format

Tagging of frames is performed for the following purposes:

a) To allow user_priority information to be added to frames carried on IEEE 802 LAN MAC types that have no inherent ability to signal priority information at the MAC protocol level;
b) To allow a frame to carry a VID;
c) To allow the frame to indicate the format of MAC Address information carried in MAC user data;
d) To allow VLANs to be supported across different MAC types.

This clause describes the tag format used for tagging frames, as follows:

e) Subclause 9.1 gives an overview of tagging;
f) Subclause 9.2 defines the data representations that are used in the descriptions of the tag field formats;
g) Subclause 9.3 describes the structure of the tag header.

Further analysis of the frame formats, the format translations that can occur when frames are tagged or untagged when relayed between different MAC methods, and a description of the tagging/untagging procedure can be found in Annex C.

The description of the tagged frame structure, both here and in Annex C, is based on two generic frame formats:

h) The frame format used in IEEE Std 802.3 MACs, and which is used with minor variations in other MACs where the native Link Layer protocol identification mechanism is based on a choice between the Type interpretation and Length interpretation of the Length/Type field. The Type interpretation is used where a Type value provides the protocol identification; the Length interpretation is used where LLC addressing provides the protocol identification. LAN MAC methods that make use of this frame format are referred to in this standard as 802.3/Ethernet MAC methods;
i) The frame format used in ISO/IEC 8802-5 and FDDI MACs, and used with minor variations in other MACs where the native Link Layer protocol identification mechanism is based on LLC addressing, and where the frame may also be able to carry source-routing information. LAN MAC methods that make use of this frame format are referred to in this standard as Token Ring/FDDI MAC methods.

For MACs other than 802.3, 8802-5, and FDDI, the approach used is to apply these frame formats, with appropriate modification to the overall frame structure, as appropriate to the MAC concerned. For example:

j) MACs such as 8802-4 and 8802-6 that use LLC as the native Link Layer protocol identification would adopt format i). If they do not provide native support for source routing, the variant of this format that is used in transparent FDDI LANs would be used;
k) MACs such as 8802-12 that can support compatibility with 802.3 and 8802-5 MACs would adopt either format h) or format i), depending upon which compatibility mode was in operation.

9.1 Overview

Tagging a frame requires

a) The addition of a tag header to the frame. This header is inserted immediately following the destination MAC Address and source MAC Address (and routing, if present) fields of the frame to be transmitted;

b) If the source and destination MAC methods differ, tagging the frame may involve translation or encapsulation of the remainder of the frame, as specified in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390;
c) Recomputation of the Frame Check Sequence (FCS).

When relaying a tagged frame between 802.3/Ethernet MACs, a Bridge may adjust the PAD field such that the minimum size of a transmitted tagged frame is 68 octets (7.2).

The tag header carries the following information:

d) The Tag Protocol Identifier (TPID) appropriate to the MAC method concerned, as described in 9.3.1. This protocol identifier identifies the frame as a tagged frame, conforming to the tagging format described in this standard.
e) Tag Control Information (TCI) as described in 9.3.2. The TCI consists of the following elements:
    1) User_priority, as described in 9.3.2.1. This field allows the tagged frame to carry user_priority information across Bridged LANs in which individual LAN segments may be unable to signal priority information (e.g., 802.3/Ethernet segments).
    2) Canonical Format Indicator (CFI), as described in 9.3.2.2. This field is used
        i) In Token Ring/source-routed FDDI MAC methods, to signal the bit order of address information carried in the encapsulated frame; and
        ii) In 802.3/Ethernet and transparent FDDI MAC methods, to signal the presence or absence of a RIF field, and, in combination with the Non-canonical Format Indicator (NCFI) carried in the RIF, to signal the bit order of address information carried in the encapsulated frame.

NOTE 1—The meaning of Canonical format as applied to MAC Addresses, and the implications of the format of addresses on the requirements for frame translation, are discussed in Annex F.

    3) VLAN Identifier (VID), as described in 9.3.2.3. This field uniquely identifies the VLAN to which the frame belongs.
f) In 802.3/Ethernet and FDDI MAC methods, an Embedded Source-Routing Information Field (E-RIF) is included, if required by the state of the CFI flag in the TCI. If present, in addition to providing the ability to carry source-routing information, this field includes a further flag, the NCFI, that signals the bit order of address information carried in the encapsulated frame. The structure of this field, as used in this context, is described in 9.3.3.

NOTE 2—The ability of the tag header to carry embedded source-routing information using 802.3/Ethernet and FDDI MAC methods does not imply a requirement on the part of a pure 802.3/Ethernet Bridge or a transparent FDDI Bridge to support source routing. This capability is provided simply to allow traffic that originates in, and is destined for, a source-routed environment to transit as VLAN-tagged traffic across a non-source-routed environment. “Tunnelling” of source-routed frames across transparent media in this manner is still required to follow the rules for source routing as defined in ISO/IEC 15802-3 (see 1.3, 5.4); in particular, Bridges that support only transparent operation are not permitted to forward any frames received that have the RII bit set in the source MAC Address field. Any use of the capability of using the E-RIF to carry real source-routing information across transparent LANs can therefore only be made by Bridges and/or end stations that support source routing. Once RIF information has been encapsulated in this way, transparent Bridges can treat the frames as transparent frames, and forward/filter them accordingly. This standard does not specify any forwarding decisions based on the E-RIF.

The structure of the tagged frame allows the following types of information to be identified and carried in tagged frames across all MAC methods:

g) Ethernet Type-encoded (E) and LLC-encoded (L) information (see 3.1, 3.2);

NOTE 3—The distinction between E and L is not represented in the tag header itself, but is identifiable by examination of the data carried in the tagged frame; in 802.3/Ethernet MAC methods, by examining the value in the Length/Type

field, and in Token Ring/FDDI MAC methods, by the presence/absence of the SNAP-based protocol identifiers used in the encapsulation formats described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390.

h) Frames in which any MAC Addresses embedded in the MAC data are carried in Canonical (C) or Non-canonical (N) format;
i) Source-routed (R) and transparent (T) frames.

NOTE 4—These abbreviations are used here and in Annex C to refer to different types of data frame. For example, a frame carrying LLC-encoded information, Non-canonical embedded addresses and source-routing information, is abbreviated to L-N-R; a frame carrying Ethernet Type-encoded information with Canonical addresses and no source-routing information would be E-C-T.

Relaying a tagged frame requires

j) If the frame formats used on the source and destination MAC methods differ, translation of the tag header to the format appropriate for the destination MAC method;
k) If the source and destination MAC methods differ, relaying the frame may require translation of the remainder of the frame, as defined in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390;
l) Inclusion/adjustment of the PAD field, if necessary, where the destination MAC method is 802.3/Ethernet (7.2);
m) Re-computation of the Frame Check Sequence (FCS) if necessary.

Untagging a tagged frame requires

n) The removal of the tag header, retaining the RIF in the appropriate position in the final untagged frame if necessary;
o) If the frame formats used on the source and destination MAC methods differ, untagging the frame may require translation of the remainder of the frame, as defined in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390;
p) Adjustment of the PAD field, if necessary, where the destination MAC method is 802.3/Ethernet (7.2);
q) Re-computation of the Frame Check Sequence (FCS).

The frame translations defined in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 are applied, as necessary, to all frames carrying Ethernet Type-encoded information relayed by VLAN-aware Bridges. Use of the CFI flag in the tagged frame allows

r) The format of embedded MAC Address information to be signalled end-to-end across a VLAN without the need for MAC Address format translation by VLAN-aware Bridges while the frame is in tagged format, regardless of the MAC methods involved in carrying the tagged frame from source to destination;

NOTE 5—In other words, from the point of view of ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 translation, the representation of Ethernet Type-encoded information is always as would be expected for the underlying MAC method, and in tagged frames, the format of embedded address information is always as indicated by the CFI/NCFI.

s) Source-routing information to be carried end-to-end across a VLAN, regardless of the MAC methods involved in carrying the tagged frame from source to destination.

The primary purpose in allowing the distinction between Canonical and Non-canonical information to be represented in the tagged frame is to permit such information to be carried across a VLAN without the need for VLAN-aware Bridges to translate the format of embedded MAC Addresses en route; however, Bridges that untag frames may still need to take the address format into consideration, and perform the appropriate translation if necessary, and if the Bridge supports such translation. For example, in order to successfully untag an L-N-T frame for transmission onto an 802.3/Ethernet segment, it would be necessary to convert any

embedded MAC Addresses to Canonical format in order for that frame to be meaningful to the end stations on that segment.

The ability to support translation of embedded MAC Addresses between Canonical and Non-canonical formats (and vice versa) when transmitting an untagged frame is not required by this standard. In Bridges that do not support such translation capability on a given outbound Port, frames that may require such translation before being forwarded as untagged frames on that Port shall be discarded.

NOTE 6—In particular, this means that a Bridge that supports only one MAC type on all Ports is not required to support MAC Address translation when untagging a frame that originated on a different MAC type.

Tagging of frames occurs when an untagged frame is relayed by a Bridge onto a LAN segment for which that frame is required by the egress rules (8.8) to be transmitted in tagged format.

Untagging of frames occurs when a tagged frame is relayed by a Bridge onto a LAN segment for which that frame is required by the egress rules (8.8) to be transmitted in untagged format.

9.2 Transmission and representation of octets

In this clause, octets in a PDU (or a field of a PDU) are numbered starting from 1 and increasing in the order in which they are put into a MAC Service Data Unit (MSDU).

The bits in an octet are numbered from 1 to 8, where 1 is the least significant bit.

Where consecutive octets are used to represent a binary number, the lower octet number carries the most significant value.

Where the value of a field is represented in hexadecimal notation, as a sequence of two-digit hexadecimal values separated by hyphens (e.g., A1-5B-03), the leftmost hexadecimal value (A1 in this example) appears in the lowest numbered octet of the field and the rightmost hexadecimal value (03 in this example) appears in the highest numbered octet of the field.

When the terms *set* and *reset* are used in the text to indicate the values of single-bit fields, *set* is encoded as a binary 1 and *reset* as a binary 0 (zero).

When the encoding of a PDU (or a field within a PDU) is represented using a diagram, the following representations are used:

a) Octets are shown with the lowest numbered octet nearest the top of the page, the octet numbering increasing from the top to bottom; or
b) Octets are shown with the lowest numbered octet nearest the left of the page, the octet numbering increasing from left to right;
c) Within an octet, bits are shown with bit 8 to the left and bit 1 to the right.

9.3 Structure of the tag header

The tag header consists of the following components:

a) The Tag Protocol Identifier (TPID) as described in 9.3.1;
b) The Tag Control Information (TCI) as described in 9.3.2;
c) In 802.3/Ethernet and non-Source-Rrouted FDDI frames (i.e., FDDI frames in which the RII bit is reset), the E-RIF, if required by the state of the CFI.

There are three forms of the tag header, depending upon the type of encoding used for the TPID and the underlying MAC type. The overall structure of the three forms of header is illustrated in Figure 9-1.

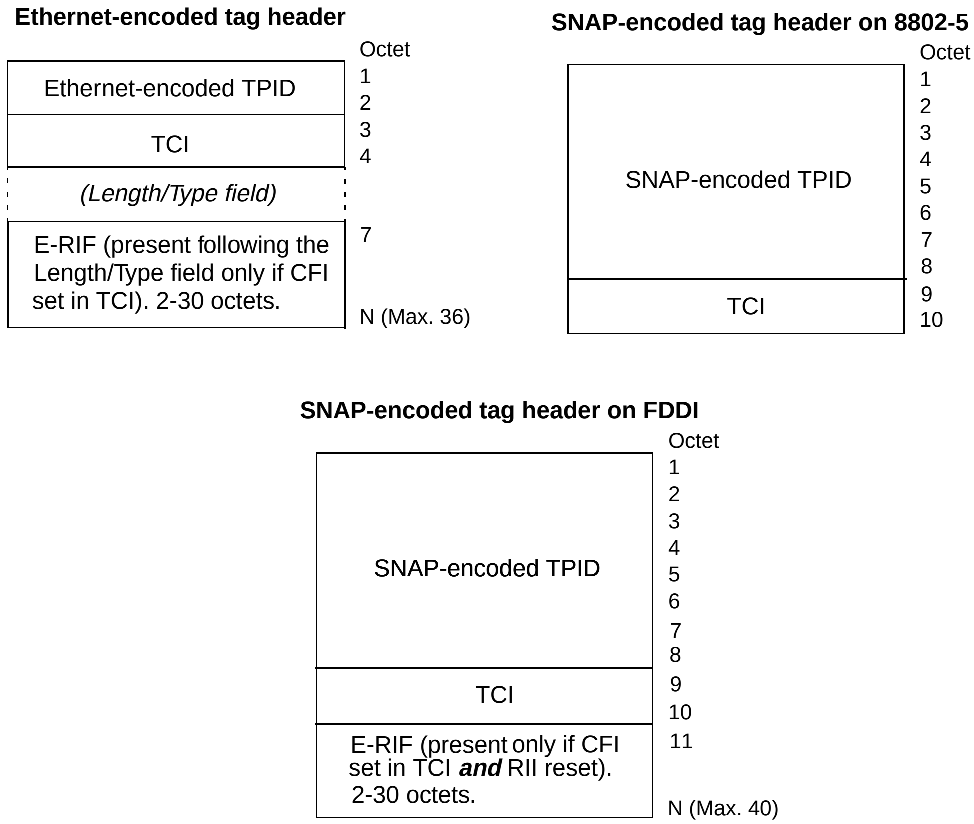


Figure 9-1—Tag header formats

The Ethernet-encoded form of the tag header is used where the tagged frame is to be transmitted using 802.3/Ethernet MAC methods.

The two SNAP-encoded forms of the tag header are used where the tagged frame is to be transmitted on Token Ring and FDDI MAC methods:

d) On 8802-5 MACs, any RIF information, if present, appears in the normal position in the frame, i.e., directly following the destination MAC Address field, and is not part on the tag header.
e) On FDDI MACs, RIF information, if present, may be carried either in the normal position in the frame (i.e., directly following the destination MAC Address field), or within the E-RIF field in the tag header.

NOTE—The SNAP-encoded frame format on FDDI allows FDDI LANs to be operated in transparent mode and/or in source routing mode.

9.3.1 Tag Protocol Identifier (TPID) format

The structure of the TPID field takes two forms, depending upon whether the field is Ethernet encoded (9.3.1.1) or SNAP encoded (9.3.1.2). The TPID carries an Ethernet Type value (802.1QTagType), which identifies the frame as a tagged frame. The value of 802.1QTagType is defined in Table 9-1.

Table 9-1—802.1Q Ethernet Type allocations

Name	Value
802.1Q Tag Protocol Type (802.1QTagType)	81-00

9.3.1.1 Ethernet-encoded TPID

The Ethernet-encoded TPID (ETPID) field is two octets in length. The ETPID carries the value of the 802.1QTagType, as defined in Table 9-1.

Figure 9-2 illustrates the structure of the ETPID.

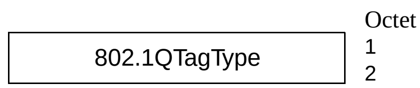


Figure 9-2—Ethernet-encoded TPID format

9.3.1.2 SNAP-encoded TPID

The SNAP-encoded TPID (STPID) is eight octets in length, encoded in SNAP format, as follows:

a) Octets numbered 1 through 3 carry the standard SNAP header, consisting of the hexadecimal value AA-AA-03;
b) Octets numbered 4 through 6 carry the SNAP PID, consisting of the hexadecimal value 00-00-00;
c) Octets 7 and 8 carry the 802.1QTagType, as defined in Table 9-1.

Figure 9-3 illustrates the structure of the STPID.

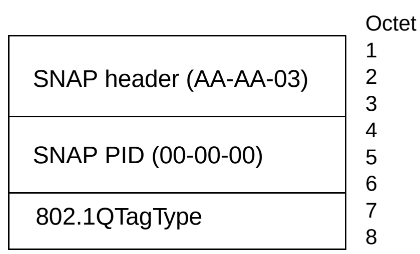


Figure 9-3—SNAP-encoded TPID format

9.3.2 Tag Control Information (TCI) format

The TCI field is two octets in length, and contains user_priority, CFI and VID (VLAN Identifier) fields. Figure 9-4 illustrates the structure of the TCI field.

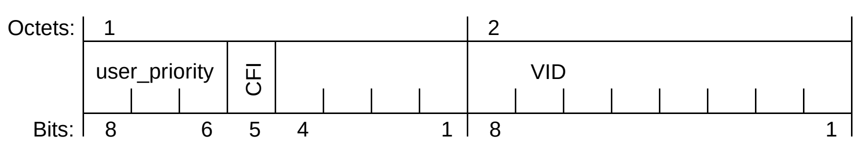


Figure 9-4—Tag Control Information (TCI) format

9.3.2.1 user_priority

The user_priority field is three bits in length, interpreted as a binary number. The user_priority is therefore capable of representing eight priority levels, 0 through 7. The use and interpretation of this field is defined in ISO/IEC 15802-3.

9.3.2.2 CFI format

The Canonical Format Indicator (CFI) is a single bit flag value. CFI reset indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format.

The meaning of the CFI when set depends upon the variant of the tag header in which it appears.

a) In a SNAP-encoded tag header transmitted using 8802-5 MAC methods, CFI has the following meanings:
   1) When set, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Non-canonical format (N);
   2) When reset, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).
b) In an Ethernet-encoded tag header, transmitted using 802.3/Ethernet MAC methods, CFI has the following meanings:
   1) When set, indicates that the E-RIF field is present in the tag header, and that the NCFI bit in the RIF determines whether MAC Address information that may be present in the MAC data carried by the frame is in Canonical (C) or Non-canonical (N) format;
   2) When reset, indicates that the E-RIF field is not present in the tag header, and that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).
c) In a SNAP-encoded tag header transmitted using FDDI MAC methods, CFI has the following meanings:
   1) When the frame takes the source-routed form (i.e., the RII bit is set in the frame's source MAC Address field and a RIF follows the source MAC Address), the interpretation of the CFI bit is as defined in a) for SNAP-encoded tag headers transmitted using 8802-5 MAC methods. The E-RIF field is not present in this form;
   2) When the frame takes the transparent form (i.e., the RII bit is reset in the frame's source MAC Address field and there is no RIF following the source MAC Address), the interpretation of the CFI bit and the presence or absence of the E-RIF is as defined in b) for Ethernet-encoded tag headers transmitted using 802.3/Ethernet MAC methods.

NOTE 1—The decision as to whether the source-routed form or the transparent form is used on FDDI is a local matter, and depends upon local knowledge in a Bridge or end station as to whether the FDDI LAN is capable of supporting source-routed traffic. The transparent form allows source-routing information to be transparently "tunneled" across LANs that do not support source routing; i.e., LANs where there may be intermediate transparent Bridges in the transmission path that would discard source-routed frames.

NOTE 2—In order to correctly relay frames between differing media types, the MAC Relay function of the Bridge needs to know the MAC type associated with each port. The means by which this information is provided to the MAC Relay function is a local matter.

9.3.2.3 VID format

The twelve-bit VLAN Identifier (VID) field uniquely identify the VLAN to which the frame belongs. The VID is encoded as an unsigned binary number. Table 9-2 identifies values of the VID field that have specific meanings or uses; the remaining values of VID are available for general use as VLAN identifiers.

A priority-tagged frame is a tagged frame whose tag header contains a VID value equal to the null VLAN ID.

NOTE 1—The specification of the ingress and egress rules for VLAN-Aware Bridges (8.6, 8.8) is such that a Bridge does not propagate priority-tagged frames; a received priority-tagged frame will acquire a VLAN classification on ingress, and will therefore either be forwarded as an untagged frame, or as a tagged frame tagged with that VLAN classification, depending upon the egress configuration for that VLAN. priority-tagged frames are therefore only ever generated by end stations.

A VLAN-tagged frame is a tagged frame whose tag header contains a VID value other than the null VLAN ID.

Table 9-2—Reserved VID values

VID value (hexadecimal)	Meaning/Use
0	The null VLAN ID. Indicates that the tag header contains only user_priority information; no VLAN identifier is present in the frame. This VID value shall not be configured as a PVID, configured in any Filtering Database entry, or used in any Management operation.
1	The default PVID value used for classifying frames on ingress through a Bridge Port. The PVID value can be changed by management on a per-Port basis.
FFF	Reserved for implementation use. This VID value shall not be configured as a PVID, configured in any Filtering Database entry, used in any Management operation, or transmitted in a tag header.

A Bridge may implement the ability to support less than the full range of VID values; i.e., for a given implementation, an upper limit, N, is defined for the VID values supported, where N is less than or equal to 4094. All implementations shall support the use of all VID values in the range 0 through their defined maximum VID, N.

NOTE 2—There is a distinction made here between the range of VID *values* (0 through N) that an implementation can support as identifiers for its active VLANs, and the maximum number of active VLANs (V) that it is able to support at any one time. An implementation that supports a maximum of, say, only 16 active VLANs (V=16) can support VIDs for those VLANs that are chosen from anywhere in the full VID number space (i.e., support N=4094), or from a subset of that number space (i.e., support N<4094). Therefore N is always greater than or equal to V.

9.3.3 Embedded RIF format

The E-RIF that can appear in Ethernet-encoded tag headers, and in the transparent form of SNAP-encoded tag headers on FDDI, is a modification of the RIF as defined in ISO/IEC 15802-3, C.3.3.2. When present, it immediately follows the Length/Type field in the 802.3/Ethernet tagged frame, or immediately follows the TCI field in an FDDI frame. It consists of two components:

a) A two-octet Route Control Field (RC);
b) Zero or more octets of Route Descriptors (up to a maximum of 28 octets), as defined by RC.

The structure and semantics associated with the Route Descriptors are as defined in ISO/IEC 15802-3, C.3.3.2.

Figure 9-5 illustrates the format of the RC component, as used in the E-RIF. The fields of the RC, and their usage, are defined in the following subclauses.

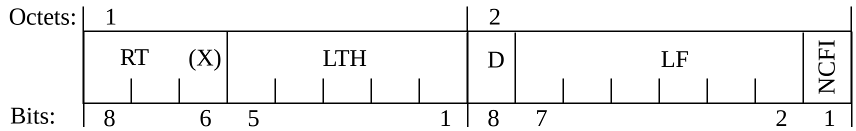


Figure 9-5—E-RIF Route Control (RC) field

Note that the definition of the E-RIF and its use within tag headers does not affect the definition of the RIF used in untagged frames in a source routing environment.

NOTE—The use of E-RIF fields in 802.3/Ethernet and FDDI frames is further discussed in 9.3.3.6.

9.3.3.1 Definition of the Routing Type (RT) field in the E-RIF

The definition of this field is as defined in ISO/IEC 15802-3, C.3.3.2, with the addition that an RT value of 01X indicates a transparent frame. The value of the rightmost bit, X, is ignored; i.e., it is the binary value 01 in bits 8 and 7 that is used to signal a transparent frame.

NOTE 1—In effect, the RT bits in the E-RIF encode the state of the RII bit that would appear in an equivalent frame in a source-routed environment. RT values 01X encode RII reset; RT values 00X or 1XX encode RII set.

The transparent frame value indicates that, with the exception of the NCFI (9.3.3.5), the remainder of the E-RIF shall be discarded if the frame is forwarded using 8802-5 MAC methods. An E-RIF containing an RT value indicating a transparent frame contains no route descriptors, and is therefore exactly 2 octets in length (however, the value of the LTH field is set to zero in this case; see 9.3.3.2).

The least significant bit of the RT field, marked (X) is reserved, as defined in ISO/IEC 15802-3, C.3.3.2.

The following rules ensure that the interpretation and use of the RT field by VLAN-aware devices is unambiguous, and does not conflict with use by non-VLAN-aware devices:

a) Where an untagged, source-routed frame is received from a Token Ring/FDDI LAN and is relayed as a tagged frame either on 802.3/Ethernet or on Token Ring/FDDI, if the received value of the RT field was 0XX, then the value of the RT field in the E-RIF or RIF in the tagged frame shall be transmitted as 000 (i.e., any 01X is converted to 000);
b) Where an untagged, transparent frame is received from a Token Ring LAN and is relayed as a tagged frame either on 802.3/Ethernet or on FDDI, then the tag header will carry an E-RIF in which the value of the RT field shall be 010;
c) Where a VLAN-aware end station on Token Ring/FDDI generates source-routed, tagged frames in the source-routed form (i.e., where the RIF appears in the normal position for a source-routed frame and there is no E-RIF in the tag header), then it shall not transmit RT values of 010 or 011 in the RIF;
d) Where a VLAN-aware end station on 802.3/Ethernet or FDDI generates source-routed, tagged frames in the transparent form (i.e., where there is source-routing information present that is carried in the E-RIF in the tag header, but there is no RIF in the normal position for a source-routed frame) then it shall not use RT values of 010 or 011 in the E-RIF;

NOTE 2—In other words, when source-routed frames are tunnelled across a transparent environment, the state of the RT bits signal that the E-RIF carries real source-routing information. An equivalent frame generated in a source-routed environment would have RII set, and the source-routing information would appear as a RIF in the normal position.

e) “X” in these rules is taken to mean “Ignored upon receipt, transmitted as zero.”

NOTE 3—The use of an RT value of 01X to indicate a transparent frame applies only to RT values carried in the E-RIF; values of 01X appearing in the RIF of a normal source-routed frame (whether tagged or untagged) are never interpreted in this way.

NOTE 4—In addition to its use as a means of tunnelling source-routed frames across transparent LANs, the above rules for the use of RT values in the E-RIF also provide the possibility of stations attached to transparent LANs using source routing to communicate with stations attached to source-routed LANs. In particular, bullet d) states the rule that makes such communication possible.

9.3.3.2 Definition of the Length (LTH) field in the E-RIF

This field is as defined in ISO/IEC 15802-3, C.3.3.2, with the exception that if the value of the RT field in the E-RIF is 01X, indicating a transparent frame, then the LTH field shall carry a value of 0.

NOTE—The use of a zero length in conjunction with the transparent RT indicator ensures that there is no possibility of such frames being misinterpreted as valid Specifically Routed frames by devices that support source routing.

9.3.3.3 Definition of the Direction Bit (D) field in the E-RIF

This field is as defined in ISO/IEC 15802-3, C.3.3.2.

9.3.3.4 Definition of the Largest Frame (LF) field in the E-RIF

This field is as defined in ISO/IEC 15802-3, C.3.3.2, and is used accordingly for all tagged frames transmitted using 802.3/Ethernet or FDDI MAC methods that carry an E-RIF, whether source-routed or transparent. For frames transmitted using 802.3/Ethernet MAC methods, the value of this field shall indicate a largest frame size of 1470 octets or less.

9.3.3.5 Definition of the NCFI field in the E-RIF

The Non-canonical Format Indicator field of the E-RIF has the following meanings:

a) When reset, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Non-canonical format (N);
b) When set, indicates that all MAC Address information that may be present in the MAC data carried by the frame is in Canonical format (C).

In source-routed frames on Token Ring/FDDI MAC methods, this bit in the RIF is reserved, and its value is preserved across Bridges; its value is normally reset.

Where a source-routed frame is received from a Token Ring/FDDI LAN and relayed as a tagged frame containing an E-RIF on 802.3/Ethernet or FDDI (i.e., where RIF information carried in the normal position for a source-routed frame is embedded in an E-RIF), the received value of this field is replaced by the appropriate N or C value.

Where a tagged frame containing an E-RIF is relayed from an 802.3/Ethernet or FDDI LAN onto a Token Ring or FDDI LAN as a source-routed frame (i.e., when the RIF information carried in the E-RIF is restored to its normal position in the frame), this bit in the RIF is reset in the frame transmitted onto the destination LAN.

9.3.3.6 E-RIF usage in tagged frames on 802.3/Ethernet and FDDI

There are three cases where a tagged frame will carry an E-RIF:

a) It is a transparent frame that carries E-N or L-N information;
b) It is a source-routed frame that carries E-N or L-N information;

c) It is a source-routed frame that carries L-C information.

Case a) occurs if

d) An untagged frame containing E-N or L-N data, and with no RIF (RII reset), is received from an 8802-5 LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet or FDDI; or
e) A tagged frame with CFI set, and with no RIF (RII reset), is received from an 8802-5 LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet or FDDI.

Case b) occurs if

f) An untagged frame containing E-N or L-N data, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing; or
g) A tagged frame with CFI set, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing.

Case c) occurs if

h) An untagged frame containing L-C data, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing; or
i) A tagged frame with CFI reset, and with a RIF (RII set), is received from a Token Ring/FDDI LAN and is transmitted as a VLAN-tagged frame on 802.3/Ethernet, or on an FDDI LAN that does not support source routing.

The other possible frame representations on Token Ring/FDDI, namely, transparent frames carrying either L-C or E-C information, are represented on 802.3/Ethernet media as VLAN-tagged frames with CFI reset (i.e., the RIF is not present).

In case a), the RIF is created from scratch as part of the frame translation. The RIF in this case consists of only 2 octets, with field values as follows:

j) RT is set to the binary value 010, to indicate a transparent frame;
k) LTH is set to 0;
l) D and LF fields are set to 0;
m) NCFI is reset to indicate that the format is Non-canonical.

In cases b) and c), the RIF contained in the frame to be translated is used unmodified, with the two exceptions that

n) Received RT values of 0XX are translated to 000 in the RT field of the E-RIF;
o) The NCFI field is set appropriately to indicate the format (Canonical or Non-canonical) of the data carried in the frame.

10. Use of GMRP in VLANs

The GARP Multicast Registration Protocol, GMRP, defined in Clause 10 of ISO/IEC 15802-3, allows the declaration and dissemination of Group membership information, in order to permit GMRP-aware Bridges to filter frames destined for group MAC Addresses on Ports through which potential recipients of such frames cannot be reached. The specification in ISO/IEC 15802-3 calls for the propagation of GMRP registrations only in the GARP Information Propagation (GIP) Context known as the *Base Spanning Tree Context* (ISO/IEC 15802-3, Clauses 10 and 12.3.4); i.e., propagation of GMRP information occurs among the set of Ports of a Bridge that are part of the *active topology* (ISO/IEC 15802-3, 7.4) of the Spanning Tree resulting from operation of the Spanning Tree Algorithm and Protocol defined in ISO/IEC 15802-3, Clause 8. This GIP Context is identified by a GIP Context Identifier of 0.

In Bridged LAN environments that support the definition and management of VLANs in accordance with this standard, the operation of GMRP as specified in ISO/IEC 15802-3 is extended to permit GMRP to operate in multiple GIP contexts, defined by the set of VLANs that are active in the Bridged LAN; these are known as *VLAN Contexts.*

The use of GMRP in a VLAN Context allows GMRP registrations to be made that are specific to that VLAN; i.e., it allows the Group filtering behavior for one VLAN to be independent of the Group filtering behavior for other VLANs. The following subclauses define the extensions to the definition of GMRP that permit its use in VLAN contexts.

With the exception of the extensions defined in this standard, the operation of GMRP and the conformance requirements associated with GMRP are as defined in ISO/IEC 15802-3.

10.1 Definition of a VLAN Context

The GIP Context Identifier used to identify a VLAN Context shall be equal to the VID used to identify the corresponding VLAN.

The set of Ports of a Bridge defined to be part of the active topology for a given VLAN Context shall be equal to the set of Ports of a Bridge for which the following are true:

a) The Port is a member of the *Member set* (8.11.9) for that VLAN; and
b) The Port is one of the Ports of the Bridge that are part of the active topology for the spanning tree that supports that VLAN.

NOTE—For the purposes of this standard, a single spanning tree is used to support all VLANs; however, the above definition has been deliberately worded so as not to preclude its use in a context where a mapping exists between M instances of spanning tree and N VLANs.

10.2 GMRP Participants and GIP Contexts

For each Port of the Bridge, a distinct instance of the GMRP Participant can exist for each VLAN Context supported by the Bridge. Each GMRP Participant maintains its own set of GARP Applicant and Registrar state machines, and its own Leave All state machine. There is no GMRP Participant associated with the Base Spanning Tree Context.

A given GARP Participant, operating in a given GIP Context, manipulates only the Port Filtering Mode and Group Registration Entry information for the context concerned. In the case of Group Registration Entries, the GIP Context Identifier value corresponds to the value of the VID field of the entry.

10.3 Context identification in GMRP PDUs

Implementations of GMRP conformant to the specification of GMRP in ISO/IEC 15802-3 exchange PDUs in the Base Spanning Tree Context; such PDUs are transmitted and received by GMRP Participants as untagged frames.

Implementations of GMRP in VLAN Bridges apply the same ingress rules (8.6) to received GMRP PDUs that are defined for the reception Port. Therefore

a) GMRP frames with no VLAN classification (i.e., untagged or priority-tagged GMRP frames) are discarded if the Acceptable Frame Types parameter (8.4.3) for the Port is set to *Admit Only VLAN-tagged frames.* Otherwise, they are classified according to the PVID for the Port;
b) VLAN-tagged GMRP frames are classified according to the VID carried in the tag header;
c) If Ingress Filtering (8.4.5) is enabled, and if the Port is not in the Member set (8.11.9) for the GMRP frame's VLAN classification, then the frame is discarded.

The VLAN classification thus associated with received GMRP PDUs establishes the VLAN Context for the received PDU, and identifies the GARP Participant instance to which the PDU is directed.

GMRP PDUs transmitted by GMRP Participants are VLAN classified according to the VLAN Context associated with that Participant. GMRP Participants in VLAN Bridges apply the same egress rules that are defined for the transmission Port (8.8). Therefore

d) GMRP PDUs are transmitted through a given Port only if the value of the Member Set for the Port for the VLAN concerned indicates that the VLAN is registered on that Port;
e) GMRP PDUs are transmitted as VLAN-tagged frames or as untagged frames in accordance with the state of the Untagged Set (8.11.9) for that Port for the VLAN concerned. Where VLAN-tagged frames are transmitted, the VID field of the tag header carries the VLAN Context Identifier value.

10.4 Default Group filtering behavior and GMRP propagation

The propagation of GMRP registrations within a VLAN context has implications with respect to the choice of default Group filtering behavior within a Bridged LAN. As GMRP frames are transmitted only on outbound Ports that are in the Member set (8.11.9) for the VLAN concerned, propagation of Group registrations by a given Bridge occurs only towards regions of the Bridged LAN where that VLAN has been (statically or dynamically) registered. This is illustrated in Figure 10-1; dotted lines in the diagram show those regions of the LAN where propagation of registrations for Group M in VLAN V does not occur. Consequently, the Filtering Databases of the lower two Bridges will not contain any Dynamic Group Registration Entry for Group M in VLAN V.

The action of these two Bridges on receipt of frames, on either of their lower Ports, destined for Group M and VLAN V, will depend upon the Default Group Filtering Behavior adopted by their upper Ports, which are the Ports that are in the Member set for VLAN V. If the Default Group Filtering Behavior is either Forward All Groups or Forward Unregistered Groups, then these Bridges will forward the frames. If the Default Group Filtering Behavior is Filter Unregistered Groups, then these Bridges will filter the frames. In the scenario shown, the choice of Default Group Filtering Behavior is therefore crucial with respect to whether or not end station S, or any other station that is "outside" the VLAN, is able to send frames to members of the Group. The choice between Filter Unregistered Groups and the other default behaviors therefore has the effect of defining VLANs that are closed to external unregistered traffic (Filter Unregistered Groups) or open to external unregistered traffic (Either of the other default behaviors).

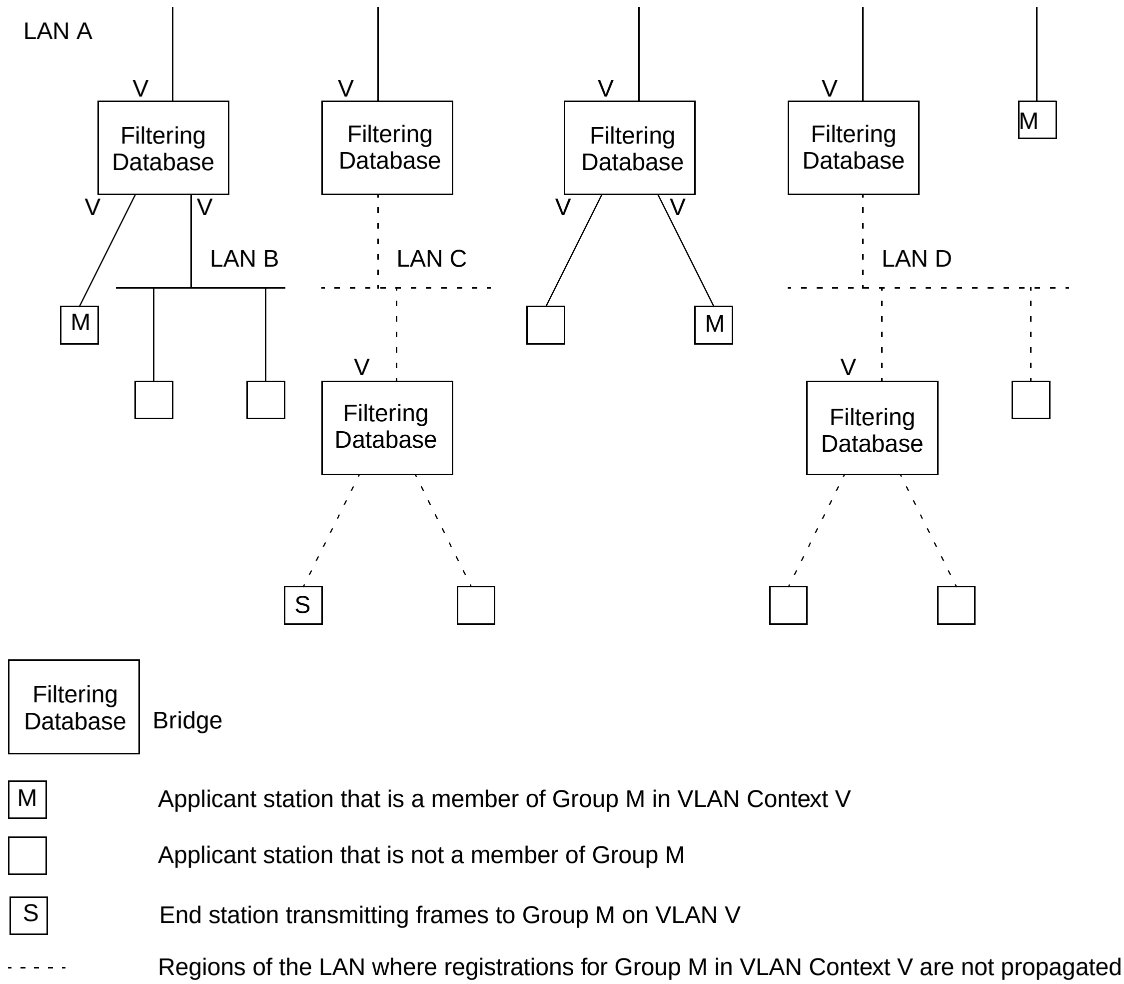


Figure 10-1—Example of GMRP propagation in a VLAN context

11. VLAN topology management

The egress rules (8.8) defined for the Forwarding Process in VLAN Bridges rely on the existence of configuration information for each VLAN that defines the set of Ports of the Bridge through which one or more members are reachable. This set of Ports is known as the Member Set (8.11.9), and its membership is determined by the presence or absence of configuration information in the Filtering Database, in the form of Static and Dynamic VLAN Registration Entries (8.11.2, 8.11.5).

Reliable operation of the VLAN infrastructure requires VLAN membership information held in the Filtering Database to be maintained in a consistent manner across all VLAN-aware Bridges in the Bridged LAN, in order to ensure that frames destined for end station(s) on a given VLAN can be correctly delivered, regardless of where in the Bridged LAN the frame is generated. Maintenance of this information by end stations that are sources of VLAN-tagged frames can allow such stations to suppress transmission of such frames if no members exist for the VLAN concerned.

This standard defines the following mechanisms that allow VLAN membership information to be configured:

a) Dynamic configuration and distribution of VLAN membership information by means of the GARP VLAN Registration Protocol (GVRP), as described in 11.2;
b) Static configuration of VLAN membership information via Management mechanisms, as described in Clause 12, which allow configuration of Static VLAN Registration Entries.

These mechanisms provide for the configuration of VLAN membership information as a result of

c) Dynamic registration actions taken by end stations or Bridges in the bridged LAN;
d) Administrative actions.

11.1 Static and dynamic VLAN configuration

The combined functionality provided by the ability to configure Static VLAN Registration Entries in the Filtering Database, coupled with the ability of GVRP to dynamically create and update Dynamic VLAN Registration Entries, offers the following possibilities with respect to how VLANs are configured on a given Port:

a) *Static configuration only.* The management facilities described in Clause 12 are used to establish precisely which VLANs have this Port in their Member set, and the GVRP management controls are used to disable the operation of the GVRP protocol on that Port. Hence, any use of GVRP by devices reachable via that Port is ignored, and the Member set for all VLANs can therefore only be determined by means of static entries in the Filtering Database.
b) *Dynamic configuration only.* The operation of GVRP is relied upon to establish Dynamic VLAN Registration Entries that will dynamically reflect which VLANs are registered on the Port, their contents changing as the configuration of the network changes. The GVRP management controls are set to enable the operation of the GVRP protocol on that Port.
c) *Combined static and dynamic configuration*. The static configuration mechanisms are used in order to configure some VLAN membership information; for other VLANs, GVRP is relied upon to establish the configuration. The GVRP management controls are set to enable the operation of the GVRP protocol on that Port.

All of the above approaches are supported by the mechanisms defined in this standard, and each approach is applicable in different circumstances. For example:

d) Use of static configuration may be appropriate on Ports where the configuration of the attached devices is fixed, or where the network administrator wishes to establish an administrative boundary

outside of which any GVRP registration information is to be ignored. For example, it might be desirable for all Ports serving end user devices to be statically configured in order to ensure that particular end users have access only to particular VLANs.

e) Use of dynamic configuration may be appropriate on Ports where the VLAN configuration is inherently dynamic; where users of particular VLANs can connect to the network via different Ports on an ad hoc basis, or where it is desirable to allow dynamic reconfiguration in the face of Spanning Tree topology changes. In particular, if the “core” of the Virtual Bridged LAN contains redundant paths that are pruned by the operation of Spanning Tree, then it is desirable for Bridge Ports that form the core network to be dynamically configured.
f) Use of both static and dynamic configuration may be appropriate on Ports where it is desirable to place restrictions on the configuration of some VLANs, while maintaining the flexibility of dynamic registration for others. For example, on Ports serving mobile end user devices, this would maintain the benefits of dynamic VLAN registration from the point of view of traffic reduction, while still allowing administrative control over access to some VLANs via that Port.

11.2 GARP VLAN Registration Protocol

The GARP VLAN Registration Protocol (GVRP) defines a *GARP Application* that provides the VLAN registration service defined in 11.2.2. GVRP makes use of GARP Information Declaration (GID) and GARP Information Propagation (GIP), which provide the common state machine descriptions and the common information propagation mechanisms defined for use in GARP-based applications. The GARP architecture, GID, and GIP are defined in ISO/IEC 15802-3, Clause 12.

GVRP provides a mechanism for dynamic maintenance of the contents of Dynamic VLAN Registration Entries for each VLAN, and for propagating the information they contain to other Bridges. This information allows GVRP-aware devices to dynamically establish and update their knowledge of the set of VLANs that currently have active members, and through which Ports those members can be reached.

11.2.1 GVRP overview

The operation of GVRP is closely similar to the operation of GMRP (ISO/IEC 15802-3, Clause 10), which is used for registering Group membership information. The primary differences are as follows:

a) The attribute values carried by the protocol are 12-bit VID values, rather than 48-bit MAC Addresses and Group service requirement information;
b) The act of registering/deregistering a VID affects the contents of Dynamic VLAN Registration Entries (8.11.5), rather than the contents of Group Registration Entries (8.11.4).

GVRP allows both end stations and Bridges in a Bridged LAN to issue and revoke declarations relating to membership of VLANs. The effect of issuing such a declaration is that each GVRP Participant that receives the declaration will create or update a Dynamic VLAN Registration Entry in the Filtering Database to indicate that VLAN is registered on the reception Port. Subsequently, if all Participants on a segment that had an interest in a given VID revoke their declarations, the Port attached to that segment is set to Unregistered in the Dynamic VLAN Registration Entry for that VLAN by each GVRP Participant attached to that segment.

Figure 11-1 illustrates the architecture of GVRP in the case of a two-Port Bridge and an end station.

As shown in the diagram, the GVRP Participant consists of the following components:

c) The GVRP Application, described in 11.2.3;
d) GARP Information Propagation (GIP), described in ISO/IEC 15802-3, Clause 12;
e) GARP Information Declaration, described in ISO/IEC 15802-3, Clause 12.

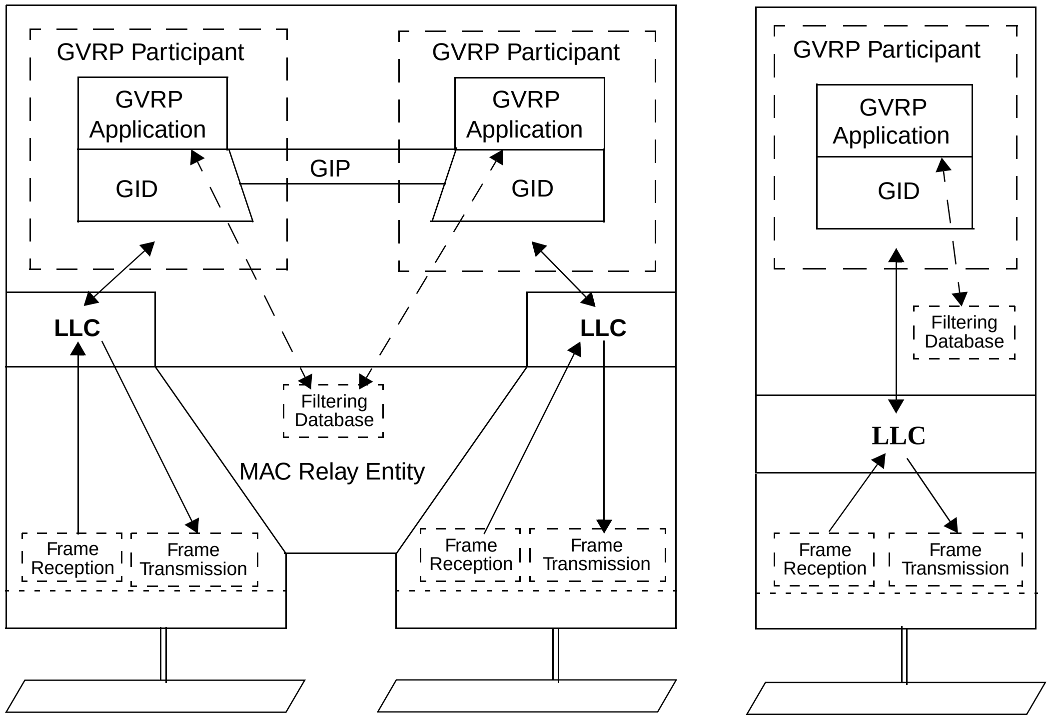


Figure 11-1—Operation of GVRP

11.2.1.1 Behavior of end stations

VLAN-aware end stations participate in GVRP protocol activity, as appropriate for the set of VLANs of which they are currently members. GVRP provides a way for such an end station to ensure that the VLAN(s) of which it is a member are registered for each Port on any LAN segment to which the end station is attached. GVRP also provides for that VID information to be propagated across the Spanning Tree to other VLAN-aware devices, as described in 11.2.1.2.

Incoming VLAN membership information (from all other devices on the same LAN segment) allows such end stations to "source prune" (i.e., discard at source; see ISO/IEC 15802-3, 10.2.2) any traffic destined for VLANs that currently have no other members in the Bridged LAN, thus avoiding the generation of unnecessary traffic on their local LAN segments. This is illustrated in Figure 11-1 by a Filtering Database shown as being present in the end station.

NOTE—Non-VLAN-aware end stations have no need to register VLAN membership via GVRP; indeed, this would be impossible for them to achieve if truly VLAN-unaware, as they would have no knowledge of the set of VLANs in which they participate. Their VLAN registration requirements are taken care of by means of the configuration of PVIDs (and possibly other VLAN classification mechanisms) and the propagation of registered VLAN IDs by the Bridges.

11.2.1.2 Behavior of Bridges

VLAN-aware Bridges register and propagate VLAN memberships on all Bridge Ports that are part of the active topology of the underlying Spanning Tree. Incoming VID registration and de-registration information is used to update the Dynamic VLAN Registration Entries associated with each VLAN. Any changes in the state of registration of a given VID on a given Port are propagated on Ports that are part of the active topol-

ogy of the Spanning Tree, in order to ensure that other GVRP-aware devices in the Bridged LAN update their Filtering Databases appropriately.

The Filtering Databases in all GVRP-aware devices are thus automatically configured such that the Port Map in the Dynamic VLAN Registration Entry for a given VID indicates that a given Port is registered if one or more members of the corresponding VLAN are reachable through the Port.

NOTE—The information that determines whether frames destined for each VLAN are transmitted VLAN-tagged or untagged is carried in Static VLAN Registration Entries (8.11.2); if no such entry exists for a VLAN, then it is assumed that frames for that VLAN are transmitted VLAN-tagged on all Ports. Therefore, if the configuration information held in the Filtering Database for a given VLAN consists only of information configured by the operation of GVRP (i.e., only a Dynamic VLAN Registration Entry), then all traffic for that VLAN will be VLAN-tagged on transmission.

11.2.1.3 Use of the PVID

The initial state of the Permanent Database contains a Static VLAN Registration Entry for the Default PVID, in which the Port Map indicates Registration Fixed on all Ports. This ensures that in the default state, where the PVID on all Ports is the Default PVID, membership of the Default PVID is propagated across the Bridged LAN to all other GVRP-aware devices. Subsequent management action may change both the Permanent Database and the Filtering Database in order to modify or remove this initial setting, and may change the PVID value on each Port of the Bridge.

NOTE—In the absence of any modification of these initial settings, this ensures that connectivity is established across the Bridged LAN for the VLAN corresponding to the Default PVID.

11.2.2 VLAN registration service definition

The VLAN registration service allows MAC Service users to indicate to the MAC Service provider the set of VLANs in which they wish to participate; i.e., that the MAC Service user wishes to receive traffic destined for members of that set of VLANs. The service primitives allow the service user to:

a) Register membership of a VLAN;
b) De-register membership of a VLAN.

Provision of these services is achieved by means of GVRP and its associated procedures, as described in 11.2.3.

ES_REGISTER_VLAN_MEMBER (VID)

Indicates to the MAC Service provider that the MAC Service user wishes to receive frames destined for the VLAN identified by the VID parameter.

ES_DEREGISTER_VLAN_MEMBER (VID)

Indicates to the MAC Service provider that the MAC Service user no longer wishes to receive frames destined for the VLAN identified by the VID parameter.

The use of these services can result in the propagation of VID information across the Spanning Tree, affecting the contents of Dynamic VLAN Registration Entries (8.11.5) in Bridges and end stations in the Bridged LAN, and thereby affecting the frame forwarding behavior of those Bridges and end stations.

11.2.3 Definition of the GVRP Application

11.2.3.1 Definition of GARP protocol elements

11.2.3.1.1 GVRP Application address

The group MAC Address used as the destination address for GARP PDUs destined for GVRP Participants shall be the GVRP address identified in Table 11-1. Received PDUs that are constructed in accordance with the PDU format defined in ISO/IEC 15802-3, 12.11, and which carry a destination MAC Address equal to the GVRP address are processed as follows:

a) In Bridges and end stations that support the operation of GVRP, all such PDUs shall be submitted to the GVRP Participant associated with the receiving Port for further processing;
b) In Bridges that do not support the operation of GVRP, all such PDUs shall be submitted to the Forwarding Process.

Table 11-1—GVRP Application address

Assignment	Value
GVRP address	01-80-C2-00-00-21

NOTE—The GVRP Application Address has been allocated from the set of GARP Application addresses defined in ISO/IEC 15802-3, Table 12-1, using the MAC Address contained in the second entry of that table.

11.2.3.1.2 Encoding of GVRP Attribute Types

The operation of GVRP defines a single Attribute Type (ISO/IEC 15802-3, 12.11.2.2) that are carried in GARP protocol exchanges; the VID Attribute Type. The VID Attribute Type is used to identify values of VLAN Identifiers (VIDs). The value of the Group Attribute Type carried in GVRP PDUs shall be 1.

11.2.3.1.3 Encoding of GVRP Attribute Values

Values of instances of the VID Attribute Type shall be encoded as Attribute Values in GARP PDUs (ISO/IEC 15802-3, 12.11.2.6) as two octets, taken to represent an unsigned binary number, and equal to the hexadecimal value of the VLAN identifier that is to be encoded.

11.2.3.2 Provision and support of the VLAN registration service

11.2.3.2.1 End system VLAN membership declaration

The GVRP Application element of a GVRP Participant provides the dynamic registration and de-registration services defined in 11.2.2, as follows:

On receipt of an ES_REGISTER_VLAN_MEMBER service primitive, the GVRP Participant issues a GID_Join.request service primitive (ISO/IEC 15802-3, 12.3.2.1). The attribute_type parameter of the request carries the value of the VID Attribute Type (11.2.3.1.2) and the attribute_value parameter carries the value of the VID parameter carried in the ES_REGISTER_VLAN_MEMBER primitive.

On receipt of an ES_DEREGISTER_VLAN_MEMBER service primitive, the GVRP Participant issues a GID_Leave.request service primitive (ISO/IEC 15802-3, 12.3.2.1). The attribute_type parameter of the

request carries the value of the VID Attribute Type (11.2.3.1.2) and the attribute_value parameter carries the value of the VID parameter carried in the ES_REGISTER_VLAN_MEMBER primitive.

11.2.3.2.2 VLAN membership registration

The GVRP Application element of a GVRP Participant responds to registration and de-registration events signalled by GID as follows:

On receipt of a GID_Join.indication (ISO/IEC 15802-3, 12.3.2.2) whose attribute_type is equal to the value of the VID Attribute Type (11.2.3.1.2), the GVRP Application element indicates the reception Port as Registered in the Port Map of the Dynamic VLAN Registration Entry for the VID indicated by the attribute_value parameter. If no such entry exists, and there is sufficient room in the Filtering Database, an entry is created.

On receipt of a GID_Leave.indication (ISO/IEC 15802-3, 12.3.2.2) whose attribute_type is equal to the value of the VID Attribute Type (11.2.3.1.2), the GVRP Application element indicates the reception Port as Unregistered in the Port Map of the Dynamic VLAN Registration Entry for the VID indicated by the attribute_value parameter. If marking this Port as Unregistered results in a Port Map that does not indicate any Port as Registered, the entry is deleted.

11.2.3.2.3 Administrative controls

The provision of static control over the declaration or registration state of the state machines associated with the GVRP Application is achieved by means of the Registrar administrative control parameters provided by GARP (ISO/IEC 15802-3, 12.9.1). These administrative control parameters are represented as Static VLAN Registration Entries in the Filtering Database (8.11.2). Where management capability is implemented, these controls can be manipulated by means of the management functionality defined in 12.7.

The provision of static control over the ability of Applicant state machines to participate in protocol exchanges is achieved by means of the Applicant Administrative Control parameters associated with the operation of GARP (ISO/IEC 15802-3, 12.9.2). Where management capability is implemented, the Applicant Administrative Control parameters can be applied and modified by means of the management functionality defined in 12.9.

11.2.3.3 GIP context for GVRP

GVRP as defined by this standard operates in the Base Spanning Tree Context (ISO/IEC 15802-3, 12.3.1); i.e., GVRP operates only on the base Spanning Tree defined by ISO/IEC 15802-3. Consequently, all GVRP PDUs sent and received by GVRP Participants are transmitted as untagged frames.

11.3 Conformance to GVRP

This subclause defines the conformance requirements for implementations claiming conformance to GVRP. Two cases are covered; implementation of GVRP in MAC Bridges and implementation of GVRP in end stations. Although this standard is principally concerned with defining the requirements for MAC Bridges, the conformance requirements for end station implementations of GVRP are included in order to give useful guidance to such implementations. The PICS proforma defined in Annex A is concerned only with conformance claims with respect to MAC Bridges.

11.3.1 Conformance to GVRP in MAC Bridges

A MAC Bridge for which conformance to GVRP is claimed shall

a) Conform to the operation of the GARP Applicant and Registrar state machines, and the LeaveAll generation mechanism, as defined in ISO/IEC 15802-3, 12.8.1, 12.8.2, and 12.8.3;
b) Exchange GARP PDUs as required by those state machines, formatted in accordance with the generic PDU format described in ISO/IEC 15802-3, 12.11, and able to carry application-specific information as defined in 11.2.3, using the GVRP Application address as defined in Table 11-1;
c) Propagate registration information in accordance with the operation of GIP for the Base Spanning Tree Context, as defined in ISO/IEC 15802-3, 12.3.3 and 12.3.4;
d) Implement the GVRP Application component as defined in 11.2;
e) Forward, filter or discard MAC frames carrying any GARP Application address as the destination MAC Address in accordance with the requirements of 8.14.3.

11.3.2 Conformance to GVRP in end stations

An end station for which conformance to GVRP is claimed shall

a) Conform to the operation of one of
   1) The Applicant state machine, as defined in ISO/IEC 15802-3, 12.8.1; or
   2) The Applicant Only state machine, as defined in ISO/IEC 15802-3, 12.8.5; or
   3) The Simple Applicant state machine, as defined in ISO/IEC 15802-3, 12.8.6;
b) Exchange GARP PDUs as required by the GARP state machine(s) implemented, formatted in accordance with the generic PDU format described in ISO/IEC 15802-3, 12.11, and able to carry application-specific information as defined in 11.2.3, using the GVRP Application address as defined in Table 11-1;
c) Support the provision of end system registration and de-registration as defined in 11.2;
d) Discard MAC frames carrying any GARP Application address as the destination MAC Address in accordance with the requirements of 8.14.3.

An end station for which conformance to GVRP is claimed may optionally

e) Conform to the operation of the GARP Registrar state machine and the LeaveAll generation mechanism, as defined in ISO/IEC 15802-3, 12.8.2 and 12.8.3; and
f) Support the provision of VLAN registration and de-registration as defined in 11.2; and
g) Filter outgoing frames destined for group MAC Addresses in accordance with registered VLAN membership information, in a manner consistent with the operation of the filtering function of the forwarding process described in 8.7.2 and the operation of the egress rules defined in 8.8.

It is recommended that only those end stations that require the ability to perform Source Pruning (11.2.1.1) conform to the operation of the Applicant state machine (ISO/IEC 15802-3, 12.8.1).

For the reasons stated in ISO/IEC 15802-3, 12.7.9, it is recommended that end stations that do not require the ability to perform Source Pruning implement the Applicant Only state machine (ISO/IEC 15802-3, 12.8.5), in preference to the Simple Applicant state machine (ISO/IEC 15802-3, 12.8.6).

NOTE—End stations that implement only a) 2) and b) through d) are equivalent to the description of the Applicant Only Participant (ISO/IEC 15802-3, 12.7.7); those that implement a) 3) and b) through d) are equivalent to the description of the Simple Applicant Participant (ISO/IEC 15802-3, 12.7.8). Such end stations require only the ability to register membership of one or more VLANs, and revoke that membership at some later point in time; for this reason, there is no requirement to support the operation of the Registrar or Leave All state machines.
End stations that implement a) 1) and b) through g) are able to perform “source pruning” as described in 11.2.1.1; i.e., to suppress the transmission of frames destined for VLANs that currently have no membership. Consequently, such end stations need to support the full Applicant state machine, in combination with the Registrar and Leave All state machines.

11.4 Procedural model

11.4.1 Purpose

This section contains an example implementation of the GVRP application defined in this Clause. This “C” code description is included in order to demonstrate the structure of the GVRP application, and to show that a reasonably low overhead implementation can be constructed. The implementation has been designed with the intent of maximizing clarity and generality, not for compactness.

The example implementation is shown in two sections:

a) Header files for the GVRP application (11.4.2);
b) The GVRP application code (11.4.3).

The example implementation also references the GARP application independent “C” code contained in ISO/IEC 15802-3:

c) Header files for the GARP application independent code (ISO/IEC 15802-3, 13.2);
d) The GARP application independent code (ISO/IEC 15802-3, 13.3).

The separation shown in the documentation of the application dependent (GVRP) and application independent (GARP) aspects of the implementation gives a clear illustration of what is involved in implementing additional applications using the same basic GARP state machines. The code is intended to be largely self-documenting, by means of in-line comments.

11.4.2 GVRP application-specific header files

11.4.2.1 gvr.h

```
/* gvr.h */
#ifndef gvr_h__
#define gvr_h__

#include “garp.h”
#include “gid.h”
#include “gip.h”

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : GARP ATTRIBUTES
 ******************************************************************************
 */

typedef unsigned Vlan_id;

typedef enum {All_attributes, Vlan_attribute}
             Attribute_type;

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : CREATION, DESTRUCTION
 ******************************************************************************
 */

extern Boolean gvr_create_gvr(int process_id, void **gvr);
   /*
    * Creates a new instance of GVR, allocating and initializing a control
    * block, returning True and a pointer to this instance if creation succeeds.
    * Also creates instances of GVD (the GARP VLAN database) and of GIP
    * (which controls information propagation).
    *
```

```
    * Ports are created by the system and added to GVR separately (see
    * gvr_added_port() and gvr_removed_port() below).
    *
    * The operating system supplied process_id is for use in subsequent calls
    * to operating system services. The system itself ensures the temporal
    * scope of process_id, guarding against timers yet to expire for destroyed
    * processes, etc. Although process_id will be implicitly supplied by many
    * if not most systems, it is made explicit in this implementation for
    * clarity.
    *
    */

extern void gvr_destroy_gvr(void *gvr);
   /*
    * Destroys an instance of GVR, destroying and deallocating the associated
    * instances of GVD and GIP, and any instances of GID remaining.
    */

extern void gvr_added_port(void *my_gvr, int port_no);
   /*
    * The system has created a new port for this application and added it to
    * the ring of GID ports. This function ensures that Static VLAN Entries
    * from the Permanent Database are represented in the GVD database (which
    * provides VLAN ID to GID index mapping) and have GID machines in the newly
    * added port (with the correct management control state). This can result
    * in the creation of new GID machines or modification of the state of
    * existing machines.
    *
    * Newly created ports are “connected” for the purpose of GARP information
    * propagation using the separate function gip_connect_port(). This should
    * be called after this function, gvr_added_port. It may cause GVRP/GIP
    * to propagate information from the static management controls through
    * other ports.
    *
    * It is assumed that new ports will be “connected” correctly before the
    * application continues as determined by the active topology of the network,
    * i.e., if stp_forwarding(port_no) gvr_connect_port(port_no);.
    *
    * As the system continues to run it should invoke gip_disconnect_port()
    * and gip_connect_port() as required to maintain the required connectivity.
    */

extern void gvr_removed_port(void *my_gvr, int port_no);
   /*
    * The system has removed and destroyed the GID port. This function should
    * provide any application-specific cleanup required.
    */

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : JOIN, LEAVE INDICATIONS
 ******************************************************************************
 */

extern void gvr_join_indication(void *my_gvr, void *my_port,
                              unsigned joining_gid_index);
   /*
    *
    */

extern void gvr_join_leave_propagated(void *my_gvr, void *my_port,
                                    unsigned gid_index);
   /*
    *
    */
```

```
extern void gvr_leave_indication(void *my_gvr, void *my_port,
                               unsigned leaving_gid_index);
   /*
    *
    */


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : PROTOCOL AND MANAGEMENT EVENTS
 ******************************************************************************
 */

extern void gvr_rcv(void *my_gvr, void *my_port, void *pdu);
   /*
    * Process an entire received pdu for this instance of GVR.
    */

extern void gvr_tx(void *my_gvr, void *my_port);
   /*
    * Transmit a pdu for this instance of GVR.
    */

#endif /* gvr_h__ */
```

11.4.2.2 gvd.h

```
/* gvd.h */
#ifndef gvd_h__
#define gvd_h__

#include “sys.h”

/******************************************************************************
 * GVD : GARP VLAN DATABASE
 ******************************************************************************
 *
 * The GARP VLAN Database maps VLAN IDs into compact GID indexes and
 * vice versa. It contains VLAN ID to index mappings for all the VLAN IDs
 * dynamically registered (except when there is a database overflow, which
 * should be an event that rarely occurs through appropriate
 * sizing) and for all those for which static controls exist.
 *
 * Taken together with the GID machines for each port (which are identified
 * by the GID indexes provided by GVD), GVD logically provides the Static VLAN
 * Registration Entries and the Dynamic VLAN Registration Entries of the abstract
 * Filtering Database (see 8.11).
 *
 * Static VLAN Registration Entries are included in this database (and have GID
 * machines defined) on an as-needed basis as ports are added to the GVR
 * Application. This example implementation assumes that the necessary information
 * is taken from Static Filtering Entries kept in a Permanent Database outside
 * the example implementation. Static VLAN Entries can also be added, changed, or
 * removed as the running system is managed.
 *
 * VLAN Registration Entries are added and removed by GVRP. Note that a
 * single VLAN ID will only give rise to one entry in this database, and one
 * GID machine per port. That machine provides the functionality for both
 * the Static VLAN Entry and the VLAN Registration Entry.
 */

extern Boolean gvd_create_gvd(int max_vlans, void **gvd);
   /*
    * Creates a new instance of gvd, allocating space for up to max_vlans
```

```
    * VLAN IDs.
    *
    * Returns True if the creation succeeded together with a pointer to the
    * gvd information.
    */

extern void gvd_destroy_gvd(void *gvd);
   /*
    * Destroys the instance of gvd, releasing previously allocated database and
    * control space.
    */

extern Boolean gvd_find_entry(  void *my_gvd, Vlan_id key,
                               unsigned *found_at_index);

extern Boolean gvd_create_entry(void *my_gvd, Vlan_id key,
                               unsigned *created_at_index);

extern Boolean gvd_delete_entry(void *my_gvd,
                               unsigned  delete_at_index);

extern Boolean gvd_get_key(     void *my_gvd, unsigned index, Vlan_id *key);

#endif /* gvd_h__ */
```

11.4.2.3 gvf.h

```
/* gvf.h */
#ifndef gvf_h__
#define gvf_h__

#include “sys.h”
#include “prw.h”
#include “gvr.h”

/******************************************************************************
 * GVF : GARP VLAN REGISTRATION APPLICATION PDU FORMATTING
 ******************************************************************************
 */

typedef struct
{  /*
    * This data structure saves the temporary state required to parse GVR
    * PDUs in particular. Gpdu provides a common basis for GARP Application
    * formatters, additional state can be added here as required by GVF.
    */

    Gpdu gpdu;

} Gvf;

typedef struct /* Gvf_msg_data */
{
    Attribute_type attribute;

    Gid_event   event;

    Vlan_id        key1;

} Gvf_msg;

extern void    gvf_rdmsg_init(Pdu *pdu, Gvf **gvf);
```



```
extern void    gvf_wrmsg_init(Gvf *gvf, Pdu  *pdu, int vlan_id);

extern Boolean gvf_rdmsg(     Gvf *gvf, Gvf_msg *msg);

extern Boolean gvf_wrmsg(     Gvf *gvf, Gvf_msg *msg);

#endif /* gvf_h__ */
```

11.4.2.4 vfdb.h

```
/* vfdb.h */
#ifndef vfdb_h__
#define vfdb_h__

#include “sys.h”

/******************************************************************************
 * VFDB : VLAN ACTIVE FILTERING DATABASE INTERFACE
 ******************************************************************************
 */

extern void vfdb_filter( int port_no, unsigned vlan_id);

extern void vfdb_forward(int port_no, unsigned vlan_id);

#endif /* vfdb_h__ */
```

11.4.3 GVRP application code

11.4.3.1 gvr.c

```
/* gvr.c */

#include “gvr.h”
#include “gid.h”
#include “gip.h”
#include “garp.h”
#include “gvd.h”
#include “gvf.h”
#include “vfdb.h”


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : IMPLEMENTATION SIZING
 ******************************************************************************
 */

enum {Max_vlans = 100};

enum {Number_of_gid_machines = Max_vlans};

enum {Unused_index = Number_of_gid_machines};

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : CREATION, DESTRUCTION
 ******************************************************************************
 */
```

```
typedef struct /* gvr */
{
    Garp      g;

    unsigned  vlan_id;

    void     *gvd; /* VLAN Registration Entry Database */

    unsigned  number_of_gvd_entries;

    unsigned  last_gvd_used_plus1;

} Gvr;


Boolean gvr_create_gvr(int process_id, void **gvr)
{  /*
    */
    Gvr *my_gvr;

    if (!sysmalloc(sizeof(Gvr), &my_gvr))
        goto gvr_creation_failure;

    my_gvr->g.process_id   = process_id;
    my_gvr->g.gid          = NULL;

    if (!gip_create_gip(Number_of_gid_machines, &my_gvr->g.gip))
        goto gip_creation_failure;

    my_gvr->g.max_gid_index = Number_of_gid_machines - 1;
    my_gvr->g.last_gid_used = Zero;

    my_gvr->g.join_indication_fn  = gvr_join_indication;
    my_gvr->g.leave_indication_fn = gvr_leave_indication;
    my_gvr->g.join_propagated_fn  = gvr_join_leave_propagated;
    my_gvr->g.leave_propagated_fn = gvr_join_leave_propagated;
    my_gvr->g.transmit_fn         = gvr_tx;
    my_gvr->g.added_port_fn       = gvr_added_port;
    my_gvr->g.removed_port_fn     = gvr_removed_port;

    if (!gvd_create_gvd(Max_vlans, &my_gvr->gvd))
        goto gvd_creation_failure;

    my_gvr->number_of_gvd_entries = Max_vlans;
    my_gvr->last_gvd_used_plus1   = 0;

    *gvr = my_gvr;    return(True);
gvd_creation_failure: gip_destroy_gip(my_gvr->g.gip);
gip_creation_failure: sysfree(my_gvr);
gvr_creation_failure: return(False);
}


void gvr_destroy_gvr(Gvr *my_gvr)
{
    Gid *my_port;

    gvd_destroy_gvd(my_gvr->gvd);
    gip_destroy_gip(my_gvr->g.gip);

    while ((my_port = my_gvr->g.gid) != NULL)
        gid_destroy_port(&my_gvr->g, my_port->port_no);
}
```



```
void gvr_added_port(Gvr *my_gvr, int port_no)
{  /*
    * Query the Permanent Database for Static VLAN Entries with “Registration
    * Forbidden” or “Registration Fixed” for this Port. Repeat the following
    * steps until there are no more entries to be found.
    *
    * Check that the VLAN ID is represented in VLD. If not, create it, then
    * create GID machines for all the other Ports with control state “Normal
    * Registration” and create the GID machine for this Port. Change the
    * control state for this Port’s GID machine to forbidden or fixed as
    * required.
    *
    */
}


void gvr_removed_port(Gvr *my_gvr, int port_no)
{  /*
    * Provide any GVR specific cleanup or management alert functions for the
    * removed Port.
    */
}


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : JOIN, LEAVE INDICATIONS
 ******************************************************************************
 */

void gvr_join_indication(Gvr *my_gvr, Gid *my_port, unsigned gid_index)
{  /*
    *
    */
    Vlan_id      key;

    gvd_get_key(my_gvr->gvd, gid_index, &key);
    vfdb_forward(my_port->port_no, key);
}


void gvr_join_leave_propagated(Gvr *my_gvr, Gid *my_port, unsigned gid_index)
{  /*
    * Nothing to be done since, unlike GMR with its Forward All Unregistered
    * Port mode, a join indication on one Port does not cause filtering to be
    * instantiated on another.
    */
}


void gvr_leave_indication(Gvr *my_gvr, Gid *my_port, unsigned leaving_gid_index)
{  /*
    *
    */
    Vlan_id  key;

    gvd_get_key(my_gvr->gvd, leaving_gid_index, &key);
    vfdb_filter(my_port->port_no, key);
}

/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : RECEIVE MESSAGE PROCESSING
 ******************************************************************************
 */
```

```
static void gvr_db_full(Gvr *my_gvr, Gid *my_port)
{  /*
    * Placeholder for management alert functions indicating registrations
    * for more VLANs have been received than can be accepted.
    */
}

static void gvr_rcv_msg(Gvr *my_gvr, Gid *my_port, Gvf_msg *msg)
{  /*
    * Process one received message.
    *
    * A LeaveAll message never causes an indication (join or leave directly),
    * even for the point-to-point link protocol enhancements (where an
    * ordinary Leave does). No further work is needed here.
    *
    * A LeaveAllRange message is currently treated exactly as a LeaveAll
    * (i.e., the range is ignored).
    *
    * All the remaining messages refer to a single attribute (i.e., a single
    * registered VLAN). Try to find a matching entry in the gvd database.
    * If one is found, dispatch the message to a routine that will
    * handle both the local GID effects and the GIP propagation to other Ports.
    *
    * If no entry is found, Leave and Empty messages can be discarded, but
    * JoinIn and JoinEmpty messages demand further treatment. First, an attempt
    * is made to create a new entry using free space (in the database, which
    * corresponds to a free GID machine set). If this fails, an attempt may be
    * made to recover space from a machine set that is in an unused or less
    * significant state. Finally, the database is considered full and the received
    * message is discarded.
    *
    * Once (if) an entry is found, Leave, Empty, JoinIn, and JoinEmpty are
    * all submitted to GID (gid_rcv_msg()), which will generate and propagate
    * Join or Leave indications as necessary.
    *
    * JoinIn and JoinEmpty may cause Join indications, which are then propagated
    * by GIP.
    *
    * On a shared medium, Leave and Empty will not give rise to indications
    * immediately. However, this routine does test for and propagate
    * Leave indications so that it can be used unchanged with a point-to-point
    * protocol enhancement.
    *
    */

    unsigned gid_index  = Unused_index;

    if (   (msg->event == Gid_rcv_leaveall)
        || (msg->event == Gid_rcv_leaveall_range))
    {
        gid_rcv_leaveall(my_port);
    }
    else
    {
        if (!gvd_find_entry(my_gvr->gvd, msg->key1, &gid_index))
        {
            if (   (msg->event == Gid_rcv_joinin)
                || (msg->event == Gid_rcv_joinempty))
            {
                if (!gvd_create_entry(my_gvr->gvd, msg->key1, &gid_index))
                {
                    if (gid_find_unused(&my_gvr->g, Zero, &gid_index))
                    {
                        gvd_delete_entry(my_gvr->gvd, gid_index);
                        (void) gvd_create_entry(my_gvr->gvd, msg->key1,
```



```
                                              &gid_index);
                   }
                   else
                       gvr_db_full(my_gvr, my_port);
        }   }   }

        if (gid_index != Unused_index)
            gid_rcv_msg(my_port, gid_index, msg->event);
}   }


void gvr_rcv(Gvr *my_gvr, Gid *my_port, Pdu *pdu)
{  /*
    * Process an entire received pdu for this instance of GVR: initialize
    * the GVF pdu parsing routine, and, while messages last, read and process
    * them one at a time.
    */
    Gvf      gvf;
    Gvf_msg  msg;

    gvf_rdmsg_init(&gvf, pdu);

    while (gvf_rdmsg(&gvf, &msg))
        gvr_rcv_msg(my_gvr, my_port, &msg);
}


/******************************************************************************
 * GVR : GARP VLAN REGISTRATION APPLICATION : TRANSMIT PROCESSING
 ******************************************************************************
 */

static void gvr_tx_msg(Gvr *my_gvr, unsigned gid_index, Gvf_msg *msg)
{  /*
    * Fill in msg fields for transmission.
    */

    if (msg->event == Gid_tx_leaveall)
    {
        msg->attribute = All_attributes;
    }
    else
    {
        msg->attribute = Vlan_attribute;

        gvd_get_key(my_gvr->gvd, gid_index, &msg->key1);
}   }


void gvr_tx(Gvr *my_gvr, Gid *my_port)
{  /*
    * Get and prepare a pdu for the transmission, if one is not available,
    * simply return; if there is more to transmit, GID will reschedule a call
    * to this function.
    *
    * Get messages to transmit from GID and pack them into the pdu using GVF
    * (GARP VLAN pdu Formatter).
    */
    Pdu         *pdu;
    Gvf          gvf;
    Gvf_msg      msg;
    Gid_event    tx_event;
    unsigned     gid_index;

    if ((tx_event = gid_next_tx(my_port, &gid_index)) != Gid_null)
```

```
    {
        if (syspdu_alloc(&pdu))
        {
            gvf_wrmsg_init(&gvf, pdu, my_gvr->vlan_id);

            do
            {
                msg.event = tx_event;

                gvr_tx_msg(my_gvr, gid_index, &msg);

                if (!gvf_wrmsg(&gvf, &msg))
                {
                    gid_untx(my_port);
                    break;
                }
            } while ((tx_event  = gid_next_tx(my_port, &gid_index))
                           != Gid_null);

            syspdu_tx(pdu, my_port->port_no);
}   }   }
```

12. VLAN Bridge Management

This clause defines the set of managed objects, and their functionality, that allow administrative configuration of VLANs.

This clause

a) Introduces the functions of management to assist in the identification of the requirements placed on Bridges for the support of management facilities.
b) Establishes the correspondence between the Processes used to model the operation of the Bridge (8.3) and the managed objects of the Bridge.
c) Specifies the management operations supported by each managed object.

12.1 Management functions

Management functions relate to the users' needs for facilities that support the planning, organization, supervision, control, protection, and security of communications resources, and account for their use. These facilities may be categorized as supporting the functional areas of Configuration, Fault, Performance, Security, and Accounting Management. Each of these is summarized in 12.1.1 through 12.1.5, together with the facilities commonly required for the management of communication resources, and the particular facilities provided in that functional area by Bridge Management.

12.1.1 Configuration Management

Configuration Management provides for the identification of communications resources, initialization, reset and close-down, the supply of operational parameters, and the establishment and discovery of the relationship between resources. The facilities provided by Bridge Management in this functional area are

a) The identification of all Bridges that together make up the Bridged LAN and their respective locations and, as a consequence of that identification, the location of specific end stations to particular individual LANs.
b) The ability to remotely reset, i.e., reinitialize, specified Bridges.
c) The ability to control the priority with which a Bridge Port transmits frames.
d) The ability to force a specific configuration of the spanning tree.
e) The ability to control the propagation of frames with specific group MAC Addresses to certain parts of the configured Bridged LAN.
f) The ability to identify the VLANs in use, and through which Ports of the Bridge frames destined for a given VLAN may be received and/or forwarded.

12.1.2 Fault Management

Fault Management provides for fault prevention, detection, diagnosis, and correction. The facilities provided by Bridge Management in this functional area are

a) The ability to identify and correct Bridge malfunctions, including error logging and reporting.

12.1.3 Performance Management

Performance Management provides for evaluation of the behavior of communications resources and of the effectiveness of communication activities. The facilities provided by Bridge Management in this functional area are

a) The ability to gather statistics relating to performance and traffic analysis. Specific metrics include network utilization, frame forward, and frame discard counts for individual Ports within a Bridge.

12.1.4 Security Management

Security Management provides for the protection of resources. Bridge Management does not provide any specific facilities in this functional area.

12.1.5 Accounting Management

Accounting Management provides for the identification and distribution of costs and the setting of charges. Bridge Management does not provide any specific facilities in this functional area.

12.2 Managed objects

Managed objects model the semantics of management operations. Operations upon an object supply information concerning, or facilitate control over, the Process or Entity associated with that object.

The managed resources of a MAC Bridge are those of the Processes and Entities established in 8.3 and ISO/IEC 15802-3, 12.2. Specifically,

a) The Bridge Management Entity (12.4 and 8.13).
b) The individual MAC Entities associated with each Bridge Port (12.5, 8.2, 8.5, and 8.9).
c) The Forwarding Process of the MAC Relay Entity (12.6, 8.2, and 8.7).
d) The Filtering Database of the MAC Relay Entity (12.7 and 8.11).
e) The Bridge Protocol Entity (12.8 and 8.12; ISO/IEC 15802-3, Clause 8).
f) GARP Participants (ISO/IEC 15802-3, Clause 12).

The management of each of these resources is described in terms of managed objects and operations below.

NOTE—The values specified in this clause, as inputs and outputs of management operations, are abstract information elements. Questions of formats or encodings are a matter for particular protocols that convey or otherwise represent this information.

12.3 Data types

This subclause specifies the semantics of operations independent of their encoding in management protocol. The data types of the parameters of operations are defined only as required for that specification.

The following data types are used:

a) Boolean.
b) Enumerated, for a collection of named values.
c) Unsigned, for all parameters specified as “the number of” some quantity, and for Spanning Tree priority values that are numerically compared. When comparing Spanning Tree priority values, the lower number represents the higher priority value.
d) MAC Address.
e) Latin1 String, as defined by ANSI X3.159, for all text strings.
f) Time Interval, an Unsigned value representing a positive integral number of seconds, for all Spanning Tree protocol timeout parameters;
g) Counter, for all parameters specified as a “count” of some quantity. A counter increments and wraps with a modulus of 2 to the power of 64.

h) GARP Time Interval, an Unsigned value representing a positive integral number of centiseconds, for all GARP protocol time-out parameters.

12.4 Bridge Management Entity

The Bridge Management Entity is described in 8.13.

The objects which comprise this managed resource are

a) The Bridge Configuration (12.4.1).
b) The Port Configuration for each Port (12.4.2).

12.4.1 Bridge Configuration

The Bridge Configuration object models the operations that modify, or enquire about, the configuration of the Bridge's resources. There is a single Bridge Configuration object per Bridge.

The management operations that can be performed on the Bridge Configuration are

a) Discover Bridge (12.4.1.1);
b) Read Bridge (12.4.1.2);
c) Set Bridge Name (12.4.1.3);
d) Reset Bridge (12.4.1.4).

12.4.1.1 Discover Bridge

12.4.1.1.1 Purpose

To solicit configuration information regarding the Bridge(s) in the Bridged LAN.

12.4.1.1.2 Inputs

a) Inclusion Range, a set of ordered pairs of specific MAC Addresses. Each pair specifies a range of MAC Addresses. A Bridge shall respond if and only if
   1) For one of the pairs, the numerical comparison of its Bridge Address with each MAC Address of the pair shows it to be greater than or equal to the first, and
   2) Less than or equal to the second, and
   3) Its Bridge Address does not appear in the Exclusion List parameter below.

The numerical comparison of one MAC Address with another, for the purpose of this operation, is achieved by deriving a number from the MAC Address according to the following procedure. The consecutive octets of the MAC Address are taken to represent a binary number; the first octet that would be transmitted on a LAN medium when the MAC Address is used in the source or destination fields of a MAC frame has the most significant value, the next octet the next most significant value. Within each octet the first bit of each octet is the least significant bit.

b) Exclusion List, a list of specific MAC Addresses.

12.4.1.1.3 Outputs

a) Bridge Address—the MAC Address for the Bridge from which the Bridge Identifier used by the Spanning Tree Algorithm and Protocol is derived (8.14.5; ISO/IEC 15802-3, 8.5.1.3).
b) Bridge Name—a text string of up to 32 characters, of locally determined significance.

c) Number of Ports—the number of Bridge Ports (MAC Entities).
d) Port Addresses—a list specifying the following for each Port:
   1) Port Number—the number of the Bridge Port (ISO/IEC 15802-3, 8.5.5.1).
   2) Port Address—the specific MAC Address of the individual MAC Entity associated with the Port (8.14.2).
e) Uptime—count in seconds of the time elapsed since the Bridge was last reset or initialized (ISO/IEC 15802-3, 8.8.1).

12.4.1.2 Read Bridge

12.4.1.2.1 Purpose

To obtain general information regarding the Bridge.

12.4.1.2.2 Inputs

None.

12.4.1.2.3 Outputs

a) Bridge Address—the MAC Address for the Bridge from which the Bridge Identifier used by the Spanning Tree Algorithm and Protocol is derived (8.14.5; ISO/IEC 15802-3, 8.5.1.3).
b) Bridge Name—a text string of up to 32 characters, of locally determined significance.
c) Number of Ports—the number of Bridge Ports (MAC Entities).
d) Port Addresses—a list specifying the following for each Port:
   1) Port Number (ISO/IEC 15802-3, 8.5.5.1).
   2) Port Address—the specific MAC Address of the individual MAC Entity associated with the Port (8.14.2).
e) Uptime—count in seconds of the time elapsed since the Bridge was last reset or initialized (ISO/IEC 15802-3, 8.8.1).

12.4.1.3 Set Bridge Name

12.4.1.3.1 Purpose

To associate a text string, readable by the Read Bridge operation, with a Bridge.

12.4.1.3.2 Inputs

a) Bridge Name—a text string of up to 32 characters.

12.4.1.3.3 Outputs

None.

12.4.1.4 Reset Bridge

12.4.1.4.1 Purpose

To reset the specified Bridge. The Filtering Database is cleared and initialized with the entries specified in the Permanent Database, and the Bridge Protocol Entity is initialized (ISO/IEC 15802-3, 8.8.1).

12.4.1.4.2 Inputs

None.

12.4.1.4.3 Outputs

None.

12.4.2 Port configuration

The Port Configuration object models the operations that modify, or inquire about, the configuration of the Ports of a Bridge. There are a fixed set of Bridge Ports per Bridge (one for each MAC interface), and each is identified by a permanently allocated Port Number.

The allocated Port Numbers are not required to be consecutive. Also, some Port Numbers may be dummy entries, with no actual LAN Port (for example, to allow for expansion of the Bridge by addition of further MAC interfaces in the future). Such dummy Ports shall support the Port Configuration management operations, and other Port-related management operations in a manner consistent with the Port being permanently disabled.

The information provided by the Port Configuration consists of summary data indicating its name and type. Specific counter information pertaining to the number of packets forwarded, filtered, and in error is maintained by the Forwarding Process resource. The management operations supported by the Bridge Protocol Entity allow for controlling the states of each Port.

The management operations that can be performed on the Port Configuration are

a) Read Port (12.4.2.1);
b) Set Port Name (12.4.2.2).

12.4.2.1 Read Port

12.4.2.1.1 Purpose

To obtain general information regarding a specific Bridge Port.

12.4.2.1.2 Inputs

a) Port Number—the number of the Bridge Port (ISO/IEC 15802-3, 8.5.5.1).

12.4.2.1.3 Outputs

a) Port Name—a text string of up to 32 characters, of locally determined significance.
b) Port Type—the MAC Entity type of the Port (IEEE Std 802.3; ISO/IEC 8802-4; ISO/IEC 8802-5; ISO/IEC 8802-6; ISO/IEC 8802-9; IEEE Std 802.9a-1995; ISO/IEC 8802-12 (IEEE Std 802.3 format); ISO/IEC 8802-12 (ISO/IEC 8802-5 format); ISO 9314; other).

12.4.2.2 Set Port Name

12.4.2.2.1 Purpose

To associate a text string, readable by the Read Port operation, with a Bridge Port.

12.4.2.2.2 Inputs

a) Port Number (ISO/IEC 15802-3, 8.5.5.1).
b) Port Name—a text string of up to 32 characters.

12.4.2.2.3 Outputs

None.

12.5 MAC entities

The Management Operations and Facilities provided by the MAC Entities are those specified in the Layer Management standards of the individual MACs. A MAC Entity is associated with each Bridge Port.

12.6 Forwarding process

The Forwarding Process contains information relating to the forwarding of frames. Counters are maintained that provide information on the number of frames forwarded, filtered, and dropped due to error. Configuration data, defining how frame priority is handled, is maintained by the Forwarding Process.

The objects that comprise this managed resource are

a) The Port Counters (12.6.1).
b) The Priority Handling objects for each Port (12.6.2);
c) The Traffic Class Table for each Port (12.6.3).

12.6.1 The Port Counters

The Port Counters object models the operations that can be performed on the Port counters of the Forwarding Process resource. There are multiple instances (one for each VLAN for each MAC Entity) of the Port Counters object per Bridge.

The management operation that can be performed on the Port Counters is Read Forwarding Port Counters (12.6.1.1).

12.6.1.1 Read forwarding port counters

12.6.1.1.1 Purpose

To read the forwarding counters associated with a specific Bridge Port.

12.6.1.1.2 Inputs

a) Port Number (ISO/IEC 15802-3, 8.5.5.1);
b) Optionally, VLAN Identifier (9.3.2.3).

If the VLAN Identifier parameter is supported, then the forwarding Port counters are maintained per VLAN per Port. If the parameter is not supported, then the forwarding Port counters are maintained per Port only.

12.6.1.1.3 Outputs

a) Frames Received—count of all valid frames received (including BPDUs, frames addressed to the Bridge as an end station and frames that were submitted to the Forwarding Process, 8.5).

b) Optionally, Octets Received—count of the total number of octets in all valid frames received (including BPDUs, frames addressed to the Bridge as an end station, and frames that were submitted to the Forwarding Process).
c) Discard Inbound—count of valid frames received that were discarded by the Forwarding Process (8.7).
d) Forward Outbound—count of frames forwarded to the associated MAC Entity (8.9).
e) Discard Lack of Buffers—count of frames that were to be transmitted through the associated Port but were discarded due to lack of buffers (8.7.3).
f) Discard Transit Delay Exceeded—count of frames that were to be transmitted but were discarded due to the maximum bridge transit delay being exceeded (buffering may have been available, 8.7.3).
g) Discard on Error—count of frames that were to be forwarded on the associated MAC but could not be transmitted (e.g., frame would be too large, ISO/IEC 15802-3, 6.3.8).
h) If Ingress Filtering is supported (8.4.5), Discard on Ingress Filtering—count of frames that were discarded as a result of Ingress Filtering being enabled.
i) Optionally, Discard on Error Details—a list of 16 elements, each containing the source address of a frame and the reason why the frame was discarded (frame too large). The list is maintained as a circular buffer. The reasons for discard on error, at present, are
    1) Transmissible service data unit size exceeded; or
    2) Discard due to Ingress Filtering. The VID associated with the last discarded frame is recorded.

12.6.2 Priority handling

The Priority Handling object models the operations that can be performed upon, or inquire about, the Default User Priority parameter, the User Priority Regeneration Table parameter, and the Outbound Access Priority Table parameter for each Port. The operations that can be performed on this object are

a) Read Port Default User Priority (12.6.2.1);
b) Set Port Default User Priority (12.6.2.2);
c) Read Port User Priority Regeneration Table (12.6.2.3);
d) Set Port User Priority Regeneration Table (12.6.2.4);
e) Read Outbound Access Priority Table (12.6.2.5).

12.6.2.1 Read Port Default User Priority

12.6.2.1.1 Purpose

To read the current state of the Default User Priority parameter (ISO/IEC 15802-3, 6.4) for a specific Bridge Port.

12.6.2.1.2 Inputs

a) Port number.

12.6.2.1.3 Outputs

a) Default User Priority value—Integer in range 0–7.

12.6.2.2 Set Port Default User Priority

12.6.2.2.1 Purpose

To set the current state of the Default User Priority parameter (ISO/IEC 15802-3, 6.4) for a specific Bridge Port.

12.6.2.2.2 Inputs

a) Port number;
b) Default User Priority value—Integer in range 0–7.

12.6.2.2.3 Outputs

None.

12.6.2.3 Read Port User Priority Regeneration Table

12.6.2.3.1 Purpose

To read the current state of the User Priority Regeneration Table parameter (8.5.1) for a specific Bridge Port.

12.6.2.3.2 Inputs

a) Port number.

12.6.2.3.3 Outputs

a) Regenerated User Priority value for Received User Priority 0—Integer in range 0–7.
b) Regenerated User Priority value for Received User Priority 1—Integer in range 0–7.
c) Regenerated User Priority value for Received User Priority 2—Integer in range 0–7.
d) Regenerated User Priority value for Received User Priority 3—Integer in range 0–7.
e) Regenerated User Priority value for Received User Priority 4—Integer in range 0–7.
f) Regenerated User Priority value for Received User Priority 5—Integer in range 0–7.
g) Regenerated User Priority value for Received User Priority 6—Integer in range 0–7.
h) Regenerated User Priority value for Received User Priority 7—Integer in range 0–7.

12.6.2.4 Set Port User Priority Regeneration Table

12.6.2.4.1 Purpose

To set the current state of the User Priority Regeneration Table parameter (8.5.1) for a specific Bridge Port.

12.6.2.4.2 Inputs

a) Port number;
b) Regenerated User Priority value for Received User Priority 0—Integer in range 0–7.
c) Regenerated User Priority value for Received User Priority 1—Integer in range 0–7.
d) Regenerated User Priority value for Received User Priority 2—Integer in range 0–7.
e) Regenerated User Priority value for Received User Priority 3—Integer in range 0–7.
f) Regenerated User Priority value for Received User Priority 4—Integer in range 0–7.
g) Regenerated User Priority value for Received User Priority 5—Integer in range 0–7.
h) Regenerated User Priority value for Received User Priority 6—Integer in range 0–7.
i) Regenerated User Priority value for Received User Priority 7—Integer in range 0–7.

12.6.2.4.3 Outputs

None.

12.6.2.5 Read Outbound Access Priority Table

12.6.2.5.1 Purpose

To read the state of the Outbound Access Priority Table parameter (Table 8-3) for a specific Bridge Port.

12.6.2.5.2 Inputs

a) Port number.

12.6.2.5.3 Outputs

a) Access Priority value for User Priority 0—Integer in range 0–7.
b) Access Priority value for User Priority 1—Integer in range 0–7.
c) Access Priority value for User Priority 2—Integer in range 0–7.
d) Access Priority value for User Priority 3—Integer in range 0–7.
e) Access Priority value for User Priority 4—Integer in range 0–7.
f) Access Priority value for User Priority 5—Integer in range 0–7.
g) Access Priority value for User Priority 6—Integer in range 0–7.
h) Access Priority value for User Priority 7—Integer in range 0–7.

12.6.3 Traffic Class Table

The Traffic Class Table object models the operations that can be performed upon, or inquire about, the current contents of the Traffic Class Table (8.7.3) for a given Port. The operations that can be performed on this object are Read Port Traffic Class Table and Set Port Traffic Class Table.

12.6.3.1 Read Port Traffic Class Table

12.6.3.1.1 Purpose

To read the contents of the Traffic Class Table (8.7.3) for a given Port.

12.6.3.1.2 Inputs

a) Port Number.

12.6.3.1.3 Outputs

a) The number of Traffic Classes, in the range 1 through 8, supported on the Port;
b) For each value of Traffic Class supported on the Port, the value of the Traffic Class in the range 0 through 7, and the set of user_priority values assigned to that Traffic Class.

12.6.3.2 Set Port Traffic Class Table

12.6.3.2.1 Purpose

To set the contents of the Traffic Class Table (8.7.3) for a given Port.

12.6.3.2.2 Inputs

a) Port number;
b) For each value of Traffic Class supported on the Port, the value of the Traffic Class in the range 0 through 7, and the set of user_priority values assigned to that Traffic Class.

NOTE—If a Traffic Class value greater than the largest Traffic Class available on the Port is specified, then the value applied to the Traffic Class Table is the largest available Traffic Class.

12.6.3.2.3 Outputs

None.

12.7 Filtering Database

The Filtering Database is described in 8.11. It contains filtering information used by the Forwarding Process (8.7) in deciding through which Ports of the Bridge frames should be forwarded.

The objects that comprise this managed resource are

a) The Filtering Database (12.7.1);
b) The Static Filtering Entries (12.7.2);
c) The Dynamic Filtering Entries (12.7.3);
d) The Group Registration Entries (12.7.4);
e) The Static VLAN Registration Entries (12.7.5);
f) The Dynamic VLAN Registration Entries (12.7.5);
g) The Permanent Database (12.7.6).

12.7.1 The Filtering Database

The Filtering Database object models the operations that can be performed on, or affect, the Filtering Database as a whole. There is a single Filtering Database object per Bridge.

The management operations that can be performed on the Database are:

a) Read Filtering Database (12.7.1.1);
b) Set Filtering Database Ageing Time (12.7.1.2);
c) Read Permanent Database (12.7.6.1);
d) Create Filtering Entry (12.7.7.1);
e) Delete Filtering Entry (12.7.7.2);
f) Read Filtering Entry (12.7.7.3);
g) Read Filtering Entry Range (12.7.7.4).

12.7.1.1 Read Filtering Database

12.7.1.1.1 Purpose

To obtain general information regarding the Bridge's Filtering Database.

12.7.1.1.2 Inputs

None.

12.7.1.1.3 Outputs

a) Filtering Database Size—the maximum number of entries that can be held in the Filtering Database.
b) Number of Static Filtering Entries—the number of Static Filtering Entries (8.11.1) currently in the Filtering Database;

c) Number of Dynamic Filtering Entries—the number of Dynamic Filtering Entries (8.11.3) currently in the Filtering Database;
d) Number of Static VLAN Registration Entries—the number of Static VLAN Registration Entries (8.11.2) currently in the Filtering Database;
e) Number of Dynamic VLAN Registration Entries—the number of Dynamic VLAN Registration Entries (8.11.5) currently in the Filtering Database.
f) Ageing Time—for ageing out Dynamic Filtering Entries when the Port associated with the entry is in the Forwarding state (8.11.3).
g) If Extended Filtering Services are supported, Number of Group Registration Entries—the number of Group Registration Entries (8.11.4) currently in the Filtering Database;

12.7.1.2 Set Filtering Database Ageing Time

12.7.1.2.1 Purpose

To set the ageing time for Dynamic Filtering Entries (8.11.3).

12.7.1.2.2 Inputs

a) Ageing Time.

12.7.1.2.3 Outputs

None.

12.7.2 A Static Filtering Entry

A Static Filtering Entry object models the operations that can be performed on a single Static Filtering Entry in the Filtering Database. The set of Static Filtering Entry objects within the Filtering Database changes only under management control.

A Static Filtering Entry object supports the following operations:

a) Create Filtering Entry (12.7.7.1);
b) Delete Filtering Entry (12.7.7.2);
c) Read Filtering Entry (12.7.7.3);
d) Read Filtering Entry Range (12.7.7.4).

12.7.3 A Dynamic Filtering Entry

A Dynamic Filtering Entry object models the operations that can be performed on a single Dynamic Filtering Entry (i.e., one that is created by the Learning Process as a result of the observation of network traffic) in the Filtering Database.

A Dynamic Filtering Entry object supports the following operations:

a) Delete Filtering Entry (12.7.7.2);
b) Read Filtering Entry (12.7.7.3);
c) Read Filtering Entry Range (12.7.7.4).

12.7.4 A Group Registration Entry

A Group Registration Entry object models the operations that can be performed on a single Group Registration Entry in the Filtering Database. The set of Group Registration Entry objects within the Filtering Database changes only as a result of GARP protocol exchanges.

A Group Registration Entry object supports the following operations:

a) Read Filtering Entry (12.7.7.3);
b) Read Filtering Entry Range (12.7.7.4).

12.7.5 A VLAN Registration Entry

A VLAN Registration Entry object models the operations that can be performed on a single VLAN Registration Entry in the Filtering Database. The set of VLAN Registration Entry objects within the Filtering Database changes under management control and also as a result of GARP protocol exchanges.

12.7.5.1 Static VLAN Registration Entry object

A Static VLAN Registration Entry object supports the following operations:

a) Create Filtering Entry (12.7.7.1);
b) Delete Filtering Entry (12.7.7.2);
c) Read Filtering Entry (12.7.7.3);
d) Read Filtering Entry Range (12.7.7.4).

12.7.5.2 Dynamic VLAN Registration Entry object

A Dynamic VLAN Registration Entry object supports the following operations:

a) Read Filtering Entry (12.7.7.3);
b) Read Filtering Entry Range (12.7.7.4).

12.7.6 Permanent Database

The Permanent Database object models the operations that can be performed on, or affect, the Permanent Database. There is a single Permanent Database per Filtering Database.

The management operations that can be performed on the Permanent Database are

a) Read Permanent Database (12.7.6.1);
b) Create Filtering Entry (12.7.7.1);
c) Delete Filtering Entry (12.7.7.2);
d) Read Filtering Entry (12.7.7.3);
e) Read Filtering Entry Range (12.7.7.4).

12.7.6.1 Read Permanent Database

12.7.6.1.1 Purpose

To obtain general information regarding the Permanent Database (8.11.10).

12.7.6.1.2 Inputs

None.

12.7.6.1.3 Outputs

a) Permanent Database Size—maximum number of entries that can be held in the Permanent Database.
b) Number of Static Filtering Entries—number of Static Filtering Entries (8.11.1) currently in the Permanent Database;
c) Number of Static VLAN Registration Entries—number of Static VLAN Registration Entries (8.11.2) currently in the Permanent Database.

12.7.7 General Filtering Database operations

In these operations on the Filtering Database, the operation parameters make use of VID values, even when operating upon a Dynamic Filtering Entry (8.11.3) whose structure carries an FID rather than a VID. In this case, the value used in the VID parameter can be any VID that has been allocated to the FID concerned (8.11.7).

12.7.7.1 Create Filtering Entry

12.7.7.1.1 Purpose

To create or update a Static Filtering Entry (8.11.1) or Static VLAN Registration Entry (8.11.2) in the Filtering Database or Permanent Database. Only static entries may be created in the Filtering Database or Permanent Database.

12.7.7.1.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.
d) Port Map—a set of control indicators, one for each Port, as specified in 8.11.1 and 8.11.2.

12.7.7.1.3 Outputs

None.

12.7.7.2 Delete Filtering Entry

12.7.7.2.1 Purpose

To delete a Filtering Entry or VLAN Registration Entry from the Filtering Database or Permanent Database.

12.7.7.2.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.

12.7.7.2.3 Outputs

None.

12.7.7.3 Read Filtering Entry

12.7.7.3.1 Purpose

To read a Filtering Entry, Group Registration Entry, or VLAN Registration Entry from the Filtering or Permanent Databases.

12.7.7.3.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
c) VID—VLAN Identifier of the entry.
d) Type—Static or Dynamic entry.

12.7.7.3.3 Outputs

a) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
b) VID—VLAN Identifier of the entry.
c) Type—Static or Dynamic entry.
d) Port Map—a set of control indicators as appropriate for the entry, as specified in 8.11.1 through 8.11.5.

12.7.7.4 Read Filtering Entry range

12.7.7.4.1 Purpose

To read a range of Filtering Database entries (of any type) from the Filtering or Permanent Databases.

Since the number of values to be returned in the requested range may have exceeded the capacity of the service data unit conveying the management response, the returned entry range is identified. The indices that define the range take on values from zero up to Filtering Database Size minus one.

12.7.7.4.2 Inputs

a) Identifier—Filtering Database or Permanent Database.
b) Start Index—inclusive starting index of the desired entry range.
c) Stop Index—inclusive ending index of the desired range.

12.7.7.4.3 Outputs

a) Start Index—inclusive starting index of the returned entry range.
b) Stop Index—inclusive ending index of the returned entry range.
c) For each index returned:
    1) Address—MAC Address of the desired entry (not present in VLAN Registration Entries).
    2) VID—VLAN Identifier of the entry.
    3) Type—Static or Dynamic entry.
    4) Port Map—a set of control indicators as appropriate for the entry, as specified in 8.11.1 through 8.11.5.

12.8 Bridge Protocol Entity

The Bridge Protocol Entity is described in 8.12 and ISO/IEC 15802-3, Clause 8.

The objects that comprise this managed resource are

a) The Protocol Entity itself.
b) The Ports under its control.

12.8.1 The Protocol Entity

The Protocol Entity object models the operations that can be performed upon, or inquire about, the operation of the Spanning Tree Algorithm and Protocol. There is a single Protocol Entity per Bridge; it can, therefore, be identified as a single fixed component of the Protocol Entity resource.

The management operations that can be performed on the Protocol Entity are

a) Read Bridge Protocol Parameters (12.8.1.1);
b) Set Bridge Protocol Parameters (12.8.1.2).

12.8.1.1 Read Bridge Protocol Parameters

12.8.1.1.1 Purpose

To obtain information regarding the Bridge's Bridge Protocol Entity.

12.8.1.1.2 Inputs

None.

12.8.1.1.3 Outputs

a) Bridge Identifier—as defined in ISO/IEC 15802-3, 8.5.3.7.
b) Time Since Topology Change—count in seconds of the time elapsed since the Topology Change flag parameter for the Bridge (ISO/IEC 15802-3, 8.5.3.12) was last True.
c) Topology Change Count—count of the times the Topology Change flag parameter for the Bridge has been set (i.e., transitioned from False to True) since the Bridge was powered on or initialized.
d) Topology Change (ISO/IEC 15802-3, 8.5.3.12).
e) Designated Root (ISO/IEC 15802-3, 8.5.3.1).
f) Root Path Cost (ISO/IEC 15802-3, 8.5.3.2).
g) Root Port (ISO/IEC 15802-3, 8.5.3.3).
h) Max Age (ISO/IEC 15802-3, 8.5.3.4).
i) Hello Time (ISO/IEC 15802-3, 8.5.3.5).
j) Forward Delay (ISO/IEC 15802-3, 8.5.3.6).
k) Bridge Max Age (ISO/IEC 15802-3, 8.5.3.7).
l) Bridge Hello Time (ISO/IEC 15802-3, 8.5.3.9).
m) Bridge Forward Delay (ISO/IEC 15802-3, 8.5.3.10).
n) Hold Time (ISO/IEC 15802-3, 8.5.3.14).

12.8.1.2 Set Bridge Protocol Parameters

12.8.1.2.1 Purpose

To modify parameters in the Bridge's Bridge Protocol Entity in order to force a configuration of the spanning tree and/or tune the reconfiguration time to suit a specific topology.

12.8.1.2.2 Inputs

a) Bridge Max Age—the new value (ISO/IEC 15802-3, 8.5.3.8).
b) Bridge Hello Time—the new value (ISO/IEC 15802-3, 8.5.3.9).
c) Bridge Forward Delay—the new value (ISO/IEC 15802-3, 8.5.3.10).
d) Bridge Priority—the new value of the priority part of the Bridge Identifier (ISO/IEC 15802-3, 8.5.3.7).

12.8.1.2.3 Outputs

None.

12.8.1.2.4 Procedure

The input parameter values are checked for compliance with ISO/IEC 15802-3, 8.10.2. If they do not comply, or the value of Bridge Max Age or Bridge Forward Delay is less than the lower limit of the range specified in ISO/IEC 15802-3, Table 8-3, then no action shall be taken for any of the supplied parameters. If the value of any of Bridge Max Age, Bridge Forward Delay, or Bridge Hello Time is outside the range specified in ISO/IEC 15802-3, Table 8-3, then the Bridge need not take action.

Otherwise, the Bridge's Bridge Max Age, Bridge Hello Time, and Bridge Forward Delay parameters are set to the supplied values. The Set Bridge Priority procedure (ISO/IEC 15802-3, 8.8.4) is used to set the priority part of the Bridge Identifier to the supplied value.

12.8.2 Bridge Port

A Bridge Port object models the operations related to an individual Bridge Port in relation to the operation of the Spanning Tree Algorithm and Protocol. There are a fixed set of Bridge Ports per Bridge; each can, therefore, be identified by a permanently allocated Port Number, as a fixed component of the Protocol Entity resource.

The management operations that can be performed on a Bridge Port are

a) Read Port Parameters (12.8.2.1);
b) Force Port State (12.8.2.2);
c) Set Port Parameters (12.8.2.3).

12.8.2.1 Read Port Parameters

12.8.2.1.1 Purpose

To obtain information regarding a specific Port within the Bridge's Bridge Protocol Entity.

12.8.2.1.2 Inputs

a) Port Number—the number of the Bridge Port.

12.8.2.1.3 Outputs

a) Uptime—count in seconds of the time elapsed since the Port was last reset or initialized (ISO/IEC 15802-3, 8.8.1).
b) State—the current state of the Port (i.e., Disabled, Listening, Learning, Forwarding, or Blocking) (ISO/IEC 15802-3, 8.4, and 8.5.5.2).

c) Port Identifier—the unique Port identifier comprising two parts, the Port Number and the Port Priority field (ISO/IEC 15802-3, 8.5.5.1).
d) Path Cost (ISO/IEC 15802-3, 8.5.5.3).
e) Designated Root (ISO/IEC 15802-3, 8.5.5.4).
f) Designated Cost (ISO/IEC 15802-3, 8.5.5.5).
g) Designated Bridge (ISO/IEC 15802-3, 8.5.5.6).
h) Designated Port (ISO/IEC 15802-3, 8.5.5.7).
i) Topology Change Acknowledge (ISO/IEC 15802-3, 8.5.5.8).

12.8.2.2 Force port state

12.8.2.2.1 Purpose

To force the specified Port into Disabled (ISO/IEC 15802-3, 8.4.5) or Blocking (ISO/IEC 15802-3, 8.4.1).

12.8.2.2.2 Inputs

a) Port Number—the number of the Bridge Port.
b) State—either Disabled or Blocking (ISO/IEC 15802-3, 8.4, 8.4.1, and 8.4.5).

12.8.2.2.3 Outputs

None.

12.8.2.2.4 Procedure

If the selected state is Disabled, the Disable Port procedure (ISO/IEC 15802-3, 8.8.3) is used for the specified Port. If the selected state is Blocking, the Enable Port procedure (ISO/IEC 15802-3, 8.8.2) is used.

12.8.2.3 Set port parameters

12.8.2.3.1 Purpose

To modify parameters for a Port in the Bridge's Bridge Protocol Entity in order to force a configuration of the spanning tree.

12.8.2.3.2 Inputs

a) Port Number—the number of the Bridge Port.
b) Path Cost—the new value (ISO/IEC 15802-3, 8.5.5.3).
c) Port Priority—the new value of the priority field for the Port Identifier (ISO/IEC 15802-3, 8.5.5.1).

12.8.2.3.3 Outputs

None.

12.8.2.3.4 Procedure

The Set Path Cost procedure (ISO/IEC 15802-3, 8.8.6) is used to set the Path Cost parameter for the specified Port. The Set Port Priority procedure (ISO/IEC 15802-3, 8.8.5) is used to set the priority part of the Port Identifier (ISO/IEC 15802-3, 8.5.5.1) to the supplied value.

12.9 GARP Entities

The operation of GARP is described in ISO/IEC 15802-3, Clause 12.

The objects that comprise this managed resource are

a) The GARP Timer objects (12.9.1);
b) The GARP Attribute Type objects (12.9.2);
c) The GARP State Machine objects (12.9.3).

12.9.1 The GARP Timer object

The GARP Timer object models the operations that can be performed upon, or inquire about, the current settings of the timers used by the GARP protocol on a given Port. The management operations that can be performed on the GARP Participant are

a) Read GARP Timers (12.9.1.1);
b) Set GARP Timers (12.9.1.2).

12.9.1.1 Read GARP Timers

12.9.1.1.1 Purpose

To read the current GARP Timers for a given Port.

12.9.1.1.2 Inputs

a) The Port identifier.

12.9.1.1.3 Outputs

a) Current value of JoinTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.1 and 12.12.1);
b) Current value of LeaveTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.2 and 12.12.1);
c) Current value of LeaveAllTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.3 and 12.12.1).

12.9.1.2 Set GARP Timers

12.9.1.2.1 Purpose

To set new values for the GARP Timers for a given Port.

12.9.1.2.2 Inputs

a) The Port identifier;
b) New value of JoinTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.1 and 12.12.1);
c) New value of LeaveTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.2 and 12.12.1);
d) New value of LeaveAllTime—Centiseconds (ISO/IEC 15802-3, 12.10.2.3 and 12.12.1).

12.9.1.2.3 Outputs

None.

12.9.2 The GARP Attribute Type object

The GARP Attribute Type object models the operations that can be performed upon, or inquire about, the operation of GARP for a given Attribute Type (ISO/IEC 15802-3, 12.11.2.2). The management operations that can be performed on a GARP Attribute Type are

a) Read GARP Applicant Controls (12.9.2.1);
b) Set GARP Applicant Controls (12.9.2.2).

12.9.2.1 Read GARP Applicant Controls

12.9.2.1.1 Purpose

To read the current values of the GARP Applicant Administrative control parameters (ISO/IEC 15802-3, 12.9.2) associated with all GARP Participants for a given Port, GARP Application and Attribute Type.

12.9.2.1.2 Inputs

a) The Port identifier;
b) The GARP Application address (ISO/IEC 15802-3, Table 12-1);
c) The Attribute Type (ISO/IEC 15802-3, 12.11.2.5).

12.9.2.1.3 Outputs

a) The current Applicant Administrative Control Value (ISO/IEC 15802-3, 12.9.2);
b) Failed Registrations—Count of the number of times that this GARP Application has failed to register an attribute of this type due to lack of space in the Filtering Database (12.10.1.6).

12.9.2.2 Set GARP Applicant Controls

12.9.2.2.1 Purpose

To set new values for the GARP Applicant Administrative control parameters (ISO/IEC 15802-3, 12.9.2) associated with all GARP Participants for a given Port, GARP Application and Attribute Type.

12.9.2.2.2 Inputs

a) The Port identifier;
b) The GARP Application address (ISO/IEC 15802-3, Table 12-1);
c) The Attribute Type (ISO/IEC 15802-3, 12.11.2.5) associated with the state machine;
d) The desired Applicant Administrative Control Value (ISO/IEC 15802-3, 12.9.2).

12.9.2.2.3 Outputs

None.

12.9.3 The GARP State Machine object

The GARP State Machine object models the operations that can be performed upon, or inquire about, the operation of GARP for a given State Machine.

The management operation that can be performed on a GARP State Machine is Read GARP State.

12.9.3.1 Read GARP State

12.9.3.1.1 Purpose

To read the current value of an instance of a GARP state machine.

12.9.3.1.2 Inputs

a) The Port identifier;
b) The GARP Application address (ISO/IEC 15802-3, Table 12-1);
c) The GIP Context (ISO/IEC 15802-3, 12.3.4);
d) The Attribute Type (ISO/IEC 15802-3, 12.11.2.2) associated with the state machine;
e) The Attribute Value (ISO/IEC 15802-3, 12.11.2.6) associated with the state machine.

12.9.3.1.3 Outputs

a) The current value of the combined Applicant and Registrar state machine for the attribute (ISO/IEC 15802-3, Table 12-6);
b) Optionally, Originator address—the MAC Address of the originator of the most recent GARP PDU that was responsible for causing a state change in this state machine (ISO/IEC 15802-3, 12.9.1).

12.10 Bridge VLAN managed objects

The following managed objects define the semantics of the management operations that can be performed upon the VLAN aspects of a Bridge:

a) The Bridge VLAN Configuration managed object (12.10.1);
b) The VLAN Configuration managed object (12.10.2);
c) The VLAN Learning Constraints managed object (12.10.3).

12.10.1 Bridge VLAN Configuration managed object

The Bridge VLAN Configuration managed object models operations that modify, or enquire about, the overall configuration of the Bridge's VLAN resources. There is a single Bridge VLAN Configuration managed object per Bridge.

The management operations that can be performed on the Bridge VLAN Configuration managed object are

a) Read Bridge VLAN Configuration (12.10.1.1);
b) Configure PVID values (12.10.1.2);
c) Configure Acceptable Frame Types parameters (12.10.1.3);
d) Configure Enable Ingress Filtering parameters (12.10.1.4);
e) Reset VLAN Bridge (12.10.1.5);
f) Notify VLAN registration failure (12.10.1.6).

12.10.1.1 Read Bridge VLAN Configuration

12.10.1.1.1 Purpose

To obtain general VLAN information from a Bridge.

12.10.1.1.2 Inputs

None.

12.10.1.1.3 Outputs

a) The 802.1Q VLAN Version number. Reported as “1” by devices that implement VLAN functionality according to this edition of the standard;
b) The optional VLAN features supported by the implementation:
    1) The maximum number of VLANs supported;
    2) Whether the implementation supports the ability to override the default PVID setting, and its egress status (VLAN-tagged or untagged) on each Port.
c) For each Port:
    1) the Port number;
    2) the PVID value (8.4.4) currently assigned to that Port;
    3) the state of the Acceptable Frame Types parameter (8.4.3). The permissible values for this parameter are:
        i) Admit only VLAN-tagged frames;
        ii) Admit all frames.
    4) the state of the Enable Ingress Filtering parameter (8.4.5); Enabled or Disabled.

12.10.1.2 Configure PVID values

12.10.1.2.1 Purpose

To configure the PVID value(s) (8.4.4) associated with one or more Ports.

12.10.1.2.2 Inputs

a) For each Port to be configured, a Port number and the PVID value to be associated with that Port.

12.10.1.2.3 Outputs

None.

12.10.1.3 Configure Acceptable Frame Types parameters

12.10.1.3.1 Purpose

To configure the Acceptable Frame Types parameter (8.4.3) associated with one or more Ports.

12.10.1.3.2 Inputs

a) For each Port to be configured, a Port number and the value of the Acceptable Frame Types parameter to be associated with that Port. The permissible values of this parameter are (as defined in 8.4.3):
    1) Admit only VLAN-tagged frames;
    2) Admit all frames.

12.10.1.3.3 Outputs

None.

12.10.1.4 Configure Enable Ingress Filtering parameters

12.10.1.4.1 Purpose

To configure the Enable Ingress Filtering parameter(s) (8.4.5) associated with one or more Ports.

12.10.1.4.2 Inputs

a) For each Port to be configured, a Port number and the value of the Enable Ingress Filtering parameter to be associated with that Port. The permissible values for the parameter are
    1) Enabled;
    2) Disabled.

12.10.1.4.3 Outputs

None.

12.10.1.5 Reset VLAN Bridge

12.10.1.5.1 Purpose

To reset all statically configured VLAN-related information in the Bridge to its default state. This operation

a) Deletes all VLAN Configuration managed objects;
b) Resets the PVID associated with each Bridge Port to the Default PVID value (Table 9-2);
c) Resets the Acceptable Frame Types parameter value associated with each Port to the default value (8.4.3).

12.10.1.5.2 Inputs

None.

12.10.1.5.3 Outputs

None.

12.10.1.6 Notify VLAN registration failure

12.10.1.6.1 Purpose

To notify a manager that GVRP (11.2.3) has failed to register a given VLAN owing to lack of resources in the Filtering Database for the creation of a Dynamic VLAN Registration Entry (8.11.5).

12.10.1.6.2 Inputs

None.

12.10.1.6.3 Outputs

a) The VID of the VLAN that GVRP failed to register;
b) The Port number of the Port on which the registration request was received.

12.10.2 VLAN Configuration managed object

The VLAN Configuration object models operations that modify, or enquire about, the configuration of a particular VLAN within a Bridge. There are multiple VLAN Configuration objects per Bridge; only one such object can exist for a given VLAN ID.

The management operations that can be performed on the VLAN Configuration are:

a) Read VLAN Configuration (12.10.2.1);
b) Create VLAN Configuration (12.10.2.2);
c) Delete VLAN Configuration (12.10.2.3);

12.10.2.1 Read VLAN Configuration

12.10.2.1.1 Purpose

To obtain general information regarding a specific VLAN Configuration.

12.10.2.1.2 Inputs

a) VLAN Identifier: a 12-bit VID.

12.10.2.1.3 Outputs

a) VLAN Name: A text string of up to 32 characters of locally determined significance;
b) List of Untagged Ports: The set of Port numbers for which this VLAN ID is a member of the Untagged set (8.11.9) for that Port;
c) List of Egress Ports: The set of Port numbers for which this VLAN ID is a member of the Member set (8.11.9) for that Port.

NOTE—The values of the Member set and the Untagged set are determined by the values held in VLAN Registration Entries in the Filtering Database (8.11.2, 8.11.5, and 8.11.9).

12.10.2.2 Create VLAN Configuration

12.10.2.2.1 Purpose

To create or update a VLAN Configuration managed object.

12.10.2.2.2 Inputs

a) VLAN Identifier: a 12-bit VID;
b) VLAN Name: A text string of up to 32 characters of locally determined significance.

NOTE—Static configuration of the Member set and the Untagged set is achieved by means of the management operations for manipulation of VLAN Registration Entries (12.7.5).

12.10.2.2.3 Outputs

None.

12.10.2.3 Delete VLAN Configuration

12.10.2.3.1 Purpose

To delete a VLAN Configuration managed object.

12.10.2.3.2 Inputs

a) VLAN Identifier: a 12-bit VID;

12.10.2.3.3 Outputs

None.

12.10.3 The VLAN Learning Constraints managed object

The VLAN Learning Constraints managed object models operations that modify, or enquire about, the set of VLAN Learning Constraints (8.11.7.2) and VID to FID allocations (8.11.7.1) that apply to the operation of the Learning Process and the Filtering Database. There is a single VLAN Learning Constraints managed object per Bridge. The object is modeled as a pair of fixed-length tables, as follows:

a) A Learning Constraint table in which each table entry either defines a single Learning Constraint or is undefined. For some of the operations that can be performed upon the table, an *entry index* is used; this identifies the number of the entry in the table, where index number 1 is the first, and N is the last (where the table contains N entries).

NOTE—The number of Learning Constraint table entries supported is an implementation option. This standard does not provide any distribution mechanism to ensure that the same set of constraints is configured in all Bridges; individual Bridges can be configured by use of the management operations defined in this subclause (for example, via the use of SNMP operating upon a VLAN Bridge MIB), but there is no in-built consistency checking to ensure that all Bridges have been provided with the same constraint information. Hence, any such consistency checking is the responsibility of the network administrator and the management applications employed in the LAN.

b) A VID to FID allocation table (8.11.7.1) with an entry per VID supported by the implementation. Each table entry indicates, for that VID, that there is currently
   1) No allocation defined; or
   2) A fixed allocation to FID X; or
   3) A dynamic allocation to FID X.

The management operations that can be performed on the VLAN Learning Constraints managed object are

c) Read VLAN Learning Constraints (12.10.3.1);
d) Read VLAN Learning Constraints for VID (12.10.3.2);
e) Set VLAN Learning Constraint (12.10.3.3);
f) Delete VLAN Learning Constraint (12.10.3.4);
g) Read VID to FID allocations (12.10.3.5);
h) Read FID allocation for VID (12.10.3.6);
i) Read VIDs allocated to FID (12.10.3.7);
j) Set VID to FID allocation (12.10.3.8);
k) Delete VID to FID allocation (12.10.3.9);
l) Notify Learning Constraint Violation (12.10.3.10);

12.10.3.1 Read VLAN Learning Constraints

12.10.3.1.1 Purpose

To read the contents of a range of one or more entries in the VLAN Learning Constraints table.

12.10.3.1.2 Inputs

a) First Entry—Entry Index of first entry to be read;
b) Last Entry—Entry Index of last entry to be read.

12.10.3.1.3 Outputs

a) List of Entries—for each entry that was read:
   1) The Entry Index;
   2) The type of the Learning Constraint: Undefined, S or I;
   3) The value of the Learning Constraint, which is one of:
      i) Undefined, indicating an empty element in the table;
      ii) An S Constraint value, consisting of a pair of VIDs;
      iii) An I Constraint value, consisting of a VID and an Independent Set Identifier.

NOTE—Where this operation is implemented using a remote management protocol, PDU size constraints may restrict the number of entries that are actually read to fewer than was requested in the input parameters. In such cases, retrieving the remainder of the desired entry range can be achieved by repeating the operation with a modified entry range specification.

12.10.3.2 Read VLAN Learning Constraints for VID

12.10.3.2.1 Purpose

To read all the VLAN Learning Constraints for a given VID.

12.10.3.2.2 Inputs

a) VID—The VLAN Identifier to which the read operation applies.

12.10.3.2.3 Outputs

a) All learning constraint values that identify the VID requested. Each value returned is either
   1) An S Constraint value, consisting of a pair of VIDs; or
   2) An I Constraint value, consisting of a VID and an Independent Set Identifier.

12.10.3.3 Set VLAN Learning Constraint

12.10.3.3.1 Purpose

To modify the contents of one of the entries in the VLAN Learning Constraints table.

12.10.3.3.2 Inputs

a) Entry Index—Entry index of the entry to be set;
b) The type of the Learning Constraint: S or I;
c) The value of the Learning Constraint, which is either:
      i) An S Constraint value, consisting of a pair of VIDs; or

    ii) An I Constraint value, consisting of a VID and an Independent Set Identifier.

12.10.3.3.3 Outputs

a) Operation status. This takes one of the following values:
    1) Operation rejected due to inconsistent learning constraint specification (8.11.7.3)—The Set operation requested setting a constraint that is inconsistent with another constraint already defined in the constraint table. The operation returns the value of the constraint concerned; or
    2) Operation rejected due to inconsistent fixed VID to FID allocation (8.11.7.3)—The Set operation requested setting a constraint that is inconsistent with a fixed VID to FID allocation already defined in the allocation table. The operation returns the value of the fixed allocation concerned; or
    3) Operation rejected due to entry index exceeding the maximum index supported by the constraint table; or
    4) Operation accepted.

12.10.3.4 Delete VLAN Learning Constraint

12.10.3.4.1 Purpose

To remove one of the entries in the VLAN Learning Constraints table. This operation has the effect of setting the value of the specified table entry to "Undefined."

12.10.3.4.2 Inputs

a) Entry Index—Entry index of the entry to be deleted.

12.10.3.4.3 Outputs

a) Operation status. This takes one of the following values:
    1) Operation rejected due to entry index exceeding the maximum index supported by the constraint table; or
    2) Operation accepted.

12.10.3.5 Read VID to FID allocations

12.10.3.5.1 Purpose

To read the contents of a range of one or more entries in the VID to FID allocation table.

12.10.3.5.2 Inputs

a) First Entry—VID of first entry to be read;
b) Last Entry—VID of last entry to be read.

12.10.3.5.3 Outputs

a) List of Entries—For each entry that was read:
    1) VID—The VLAN Identifier for this entry;
    2) Allocation Type—The type of the allocation: Undefined, Fixed or Dynamic;
    3) FID—The FID to which the VID is allocated (if not of type Undefined).

NOTE—Where this operation is implemented using a remote management protocol, PDU size constraints may restrict the number of entries that are actually read to fewer than was requested in the input parameters. In such cases, retrieving

the remainder of the desired entry range can be achieved by repeating the operation with a modified entry range specification.

12.10.3.6 Read FID allocation for VID

12.10.3.6.1 Purpose

To read the FID to which a specified VID is currently allocated.

12.10.3.6.2 Inputs

a) VID—The VLAN Identifier to which the read operation applies.

12.10.3.6.3 Outputs

a) VID—the VLAN Identifier to which the read operation applies;
b) Allocation Type—the type of the allocation: Undefined, Fixed or Dynamic;
c) FID—the FID to which the VID is allocated (if not of type Undefined).

12.10.3.7 Read VIDs allocated to FID

12.10.3.7.1 Purpose

To read all the VIDs currently allocated to a given FID.

12.10.3.7.2 Inputs

a) FID—the Filtering Identifier to which the read operation applies.

12.10.3.7.3 Outputs

a) FID—the Filtering Identifier to which the read operation applies
b) Allocation List—a list of allocations for this FID. For each element in the list:
   1) Allocation Type—the type of the allocation: Fixed or Dynamic;
   2) VID—the VID that is allocated.

12.10.3.8 Set VID to FID allocation

12.10.3.8.1 Purpose

To establish a fixed allocation of a VID to an FID.

12.10.3.8.2 Inputs

a) VID—the VID of the entry to be set;
b) FID—the FID to which the VID is to be allocated.

12.10.3.8.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to inconsistent learning constraint specification (8.11.7.3)—The Set operation requested setting a fixed allocation that is inconsistent with a VLAN Learning Constraint. The operation returns the value of the VLAN Learning Constraint concerned; or

2) Operation rejected due to VID exceeding the maximum VID supported by the allocation table; or
3) Operation rejected due to FID exceeding the maximum ID supported by the implementation; or
4) Operation accepted.

12.10.3.9 Delete VID to FID allocation

12.10.3.9.1 Purpose

To remove a fixed VID to FID allocation from the VID to FID allocation table. This operation has the effect of setting the value of the specified table entry to "Undefined."

NOTE—If the VID concerned represents a currently active VLAN, then removal of a fixed allocation may result in the "Undefined" value in the table immediately being replaced by a dynamic allocation to an FID.

12.10.3.9.2 Inputs

a) VID—VID of the allocation to be deleted.

12.10.3.9.3 Outputs

a) Operation status. This takes one of the following values:
   1) Operation rejected due to VID exceeding the maximum value supported by the allocation table; or
   2) Operation accepted.

12.10.3.10 Notify Learning Constraint Violation

12.10.3.10.1 Purpose

To alert the Manager to the existence of a Learning Constraint violation (8.11.7.3). This is an unsolicited notification from the management entity of the Bridge, issued upon detection of the constraint violation.

NOTE—As indicated in 8.11.7.3, a single change in configuration, such as the registration of a new VID by GVRP or the addition of a new learning constraint, can give rise to more than one violation being notified, depending upon the set of learning constraints currently configured in the Bridge.

12.10.3.10.2 Inputs

a) None.

12.10.3.10.3 Outputs

a) Violation Type/Argument—one of
   1) Shared VLAN Learning not supported. The argument returned indicates the VIDs of a pair of active VLANs for which an S constraint exists.
   2) Independent VLAN Learning not supported. The argument returned indicates the VIDs of a pair of active VLANs for which I constraints exist that contain the same independent set identifier.
   3) Required FID range not supported. The argument returned indicates
      i) The VID that the Bridge is unable to allocate to an FID;
      ii) The maximum number of FIDs supported by the Bridge.

The violation type *Required FID range not supported* is detected only by IVL or IVL/SVL Bridges that support fewer than 4094 FIDs.

Annex A

(normative)

PICS proforma[1]

A.1 Introduction

The supplier of a protocol implementation which is claimed to conform to this standard shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight;
b) By the supplier and acquirer—or potential acquirer—of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;
c) By the user—or potential user—of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICSs);
d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

A.2 Abbreviations and special symbols

A.2.1 Status symbols

M mandatory
O optional
O.n optional, but support of at least one of the group of options labelled by the same numeral n is required
X prohibited
pred: conditional-item symbol, including predicate identification: see A.3.4
¬ logical negation, applied to a conditional item’s predicate

A.2.2 General abbreviations

N/A not applicable
PICS Protocol Implementation Conformance Statement

[1]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

A.3 Instructions for completing the PICS proforma

A.3.1 General structure of the PICS proforma

The first part of the PICS proforma, implementation identification and protocol summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire, divided into several subclauses, each containing a number of individual items. Answers to the questionnaire items are to be provided in the right-most column, either by simply marking an answer to indicate a restricted choice (usually Yes or No), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column. The second column contains the question to be answered; the third column records the status of the item—whether support is mandatory, optional, or conditional: see also A.3.4 below. The fourth column contains the reference or references to the material that specifies the item in the main body of this standard, and the fifth column provides the space for the answers.

A supplier may also provide (or be required to provide) further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labelled *Ai* or *Xi*, respectively, for cross-referencing purposes, where *i* is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format and presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformation Statement for the implementation in question.

NOTE—Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, in case that makes for easier and clearer presentation of the information.

A.3.2 Additional information

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and a PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations, or information about aspects of the implementation that are outside the scope of this standard but that have a bearing upon the answers to some items.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

A.3.3 Exception information

It may occasionally happen that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this: instead, the supplier shall write the missing answer into the Support column, together with an *Xi* reference to an item of Exception Information, and shall provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard.

NOTE—A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

A.3.4 Conditional status

A.3.4.1 Conditional items

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory or optional—are dependent upon whether or not certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the “Not Applicable” answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form “**pred:** S” where **pred** is a predicate as described in A.3.4.2 below, and S is a status symbol, M or 0.

If the value of the predicate is true (see A.3.4.2), the conditional item is applicable, and its status is indicated by the status symbol following the predicate: the answer column is to be marked in the usual way. If the value of the predicate is false, the “Not Applicable” (N/A) answer is to be marked.

A.3.4.2 Predicates

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: the value of the predicate is true if the item is marked as supported, and is false otherwise;

b) A predicate-name, for a predicate defined as a boolean expression constructed by combining item-references using the boolean operator OR: the value of the predicate is true if one or more of the items is marked as supported;

c) The logical negation symbol “¬” prefixed to an item-reference or predicate-name: the value of the predicate is true if the value of the predicate formed by omitting the “¬” symbol is false, and vice versa.

Each item whose reference is used in a predicate or predicate definition, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

A.3.4.3 References to the text of ISO/IEC 15802-3

Many of the tables in the PICS Proforma refer to the text of ISO/IEC 15802-3 (ANSI/IEEE Std 802.1D). A short form reference, of the form {D}X, is used in the “References” columns of these tables to denote references to clauses, subclauses or tables in ISO/IEC 15802-3, where X is the clause, subclause or table identifier.

A.4 PICS proforma for IEEE Std 802.1Q-1998

A.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification - e.g., name(s) and version(s) of machines and/or operating system names	

NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirement for full identification.
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).

A.4.2 Protocol summary, IEEE Std 802.1Q-1998

Identification of protocol specification	IEEE Std 802.1Q-1998, IEEE Standards for Local and Metropolitan Area Networks: Standard for Virtual Bridged Local Area Networks
Identification of amendments and corrigenda to the PICS proforma which have been completed as part of the PICS	Amd. : Corr. : Amd. : Corr. :
Have any Exception items been required? (See A.3.3: the answer Yes means that the implementation does not conform to IEEE Std 802.1Q-1998)	No [ ] Yes [ ]

Date of Statement	

A.5 Major capabilities and options

Item	Feature	Status	References	Support
(1a)*	Communications Support Which MAC types are supported on Bridge Ports, implemented in conformance with the relevant MAC standards?		{D}6.5	
(1a.1)*	CSMA/CD, IEEE Std 802.3	O.1		Yes [ ] No [ ]
(1a.2)*	Token Bus, ISO/IEC 8802-4	O.1		Yes [ ] No [ ]
(1a.3)*	Token Ring, ISO/IEC 8802-5	O.1		Yes [ ] No [ ]
(1a.4)*	FDDI, ISO 9314-2	O.1		Yes [ ] No [ ]
(1a.5)*	DQDB, ISO/IEC 8802-6	O.1		Yes [ ] No [ ]
(1a.6)*	ISLAN, ISO/IEC 8802-9	O.1		Yes [ ] No [ ]
(1a.7)*	ISLAN 16-T, IEEE 802.9a	O.1		Yes [ ] No [ ]
(1a.8)*	Demand Priority, ISO/IEC 8802-12 (IEEE Std 802.3 format)	O.1		Yes [ ] No [ ]
(1a.9)*	Demand Priority, ISO/IEC 8802-12 (ISO/IEC 8802-5 format)	O.1		Yes [ ] No [ ]
(1b)	Is LLC Type 1 supported on all Bridge Ports in conformance with ISO/IEC 8802-2?	M	8.2, 8.3, 8.14, ISO/IEC 8802-2	Yes [ ]
(1c)*	Is Source-Routing Transparent Bridge operation supported on any of the Bridge Ports? (If support is claimed, the PICS proforma detailed in ISO/IEC 15802-3, Annex D, shall also be completed).	O	{D}Annex C	Yes [ ] No [ ]
(2)	Relay and filtering of frames (A.6)	M	8.5, 8.9, 8.6, 8.7, 8.8	Yes [ ]
(2a)	Does the Bridge support Basic Filtering Services?	M	{D}6.6.5, 8.7.2	Yes [ ]
(2b)*	Does the Bridge support Extended Filtering Services?	O	{D}6.6.5, 8.7.2	Yes [ ] No [ ]
	If item (2b) is not supported, mark “N/A” and continue at (2e).			N/A[ ]
(2c)*	Does the Bridge support dynamic Group forwarding and filtering behavior?	2b:M	{D}6.6.5	Yes [ ] No [ ]
(2d)	Does the Bridge support the ability for static filtering information for individual MAC Addresses to specify a subset of Ports for which forwarding or filtering decisions are taken on the basis of dynamic filtering information?	2b:O	{D}6.6.5	Yes [ ] No [ ]
(2e)*	Does the Bridge support expedited traffic classes on any of its Ports?	O	8.1.2, 8.7.3	Yes [ ] No [ ]
(4)*	Does the Bridge support management of the priority of relayed frames?	O	{D}6.5, 8.5.1, 8.7.3, 8.7.5, Table 8-1, Table 8-2, Table 8-3	Yes [ ] No [ ]
(5)	Maintenance of filtering information (A.7)	M	8.10, 8.11	Yes [ ]
(7a)	Can the Filtering Database be read by management?	O	8.11	Yes [ ] No [ ]

A.5 Major capabilities and options *(Continued)*

Item	Feature	Status	References	Support
(7c)*	Can Static Filtering Entries be created and deleted?	O	8.11.1	Yes [ ] No [ ]
(7g)	Can Static Filtering Entries be created and deleted in the Permanent Database?	O	8.11.10	Yes [ ] No [ ]
(7h)	Can Static Filtering Entries be created for a given MAC Address specification with a distinct Port Map for each inbound Port?	O	8.11.1	Yes [ ] No [ ]
(7i)	Can Group Registration Entries be dynamically created, updated and deleted by GMRP?	2c:M	8.11.4, {D}10	Yes [ ] N/A [ ]
(10)	Addressing (A.8)	M	8.14	Yes [ ]
(9a)*	Can the Bridge be configured to use 48-bit Universal Addresses?	O.3	8.14	Yes [ ] No [ ]
(9b)*	Can the Bridge be configured to use 48-bit Local Addresses?	O.3	8.14	Yes [ ] No [ ]
(13)*	Spanning Tree algorithm and protocol (A.9)	M	{D}8, {D}9	Yes [ ]
(16)*	Does the Bridge support management of the Spanning Tree topology?	O	{D}8.2	Yes [ ] No [ ]
(17)*	Does the Bridge support management of the protocol timers?	O	{D}8.10	Yes [ ] No [ ]
(19)*	VLAN Bridge Management Operations	O	12	Yes [ ] No [ ]
(20a)*	Are the Bridge Management Operations supported via a Remote Management Protocol?	19:O.4	{D}5	Yes [ ] No [ ] N/A [ ]
(20b)*	Are the Bridge Management Operations supported via a local management interface?	19:O.4	{D}5	Yes [ ] No [ ] N/A [ ]
(23a)*	Does the implementation support, on each Port, one or more of the permissible combinations of values for the Acceptable Frame Types parameter?	M	5.1, 8.4.3	Yes [ ]
(23a.1)	State which Ports support: — Admit only VLAN-tagged frames; — Admit all frames.	M	5.1, 8.4.3	 Ports: _____ Ports: _____
(23a.2)	On Ports that support both values, is the parameter configurable via management?	M	5.1, 8.4.3, 12.10	Yes [ ] N/A [ ]
(23b)	Does the implementation support the ability to insert tag headers into, modify tag headers in, and remove tag headers from relayed frames, as required by the capabilities of each Bridge Port?	M	5.1, 7.1, 9	Yes [ ]
(23c)	Does the implementation support the ability to perform automatic configuration and management of VLAN topology information by means of GVRP on all Ports?	M	5.1, 11	Yes [ ]

A.5 Major capabilities and options *(Continued)*

Item	Feature	Status	References	Support
(23d)	Does the implementation support the ability for the Filtering Database to contain static and dynamic configuration information for at least one VLAN, by means of Static and Dynamic VLAN Registration Entries?	M	5.1, 8.11	Yes [ ]
(23d.1)	State the maximum number of VLANs supported by the implementation.	M	5.1, 8.11, 9.3.2.3	_____ VLANs
(23d.2)	State the range of VID values supported by the implementation.	M	8.11, 9.3.2.3	0 through _____
(23e)*	VLAN Learning support		5.1, 8.11.3, 8.11.7, 8.11.8	
(23e.1)	Does the implementation support at least one FID?	M		Yes [ ]
(23e.2)	Can the implementation allocate at least one VID to each FID supported?	M		Yes [ ]
(23e.4)	State the maximum number of FIDs that can be supported by the implementation.	M	8.11.7	____ FIDs
(23e.5)	State the maximum number of VIDs that can be allocated to each FID.	M	8.11.7	____ VIDs
(23e.6)	Does the implementation support configuration of VLAN Learning Constraints via management?	O	5.2, 8.11.7, 12.10.3	Yes [ ] No [ ]
(23e.7)	State the number of VLAN Learning Constraints that can be configured in the implementation.	23e.6:M	5.2, 8.11.7, 12.10.3	____ Constraints
(23e.8)	Does the implementation support configuration of VID to FID allocations via management?	O	5.2, 8.11.7.1, 12.10.3	Yes [ ] No [ ]
(23e.9)	Does the implementation take account of the allocation of VIDs to FIDs when making forwarding decisions relative to group MAC Addresses?	O	8.11.8	Yes [ ] No [ ]
(23f)	On Ports that support untagged and priority-tagged frames, does the implementation support:		5.1, 8.4.4, 8.11.9, 12.10	
(23f.1)	— A PVID value?	M		Yes [ ] N/A [ ]
(23f.2)	— The ability to configure one VLAN whose Untagged set includes that Port?	M		Yes [ ] N/A [ ]
(23f.3)	— Configuration of the PVID value via management operations?	M		Yes [ ] N/A [ ]
(23f.4)	— Configuration of Static Filtering Entries via management operations?	M		Yes [ ] N/A [ ]
(23f.5)	— The ability to configure more than one VLAN whose Untagged set includes that Port?	O		Yes [ ] No [ ] N/A [ ]
(23g)*	Does the implementation support the ability to enable and disable Ingress Filtering?	O	5.2, 8.4.5	

A.5 Major capabilities and options *(Continued)*

Item	Feature	Status	References	Support
(23h)	Does the implementation support VLAN management operations?	19:O	5.2, 12.10.2, 12.10.3	Yes [ ] No [ ]
(23i)	Is the minimum tagged frame length that can be transmitted on IEEE Std 802.3 Ports less than 68 (but 64 or more) octets?	1a.1:O	7.2	Yes [ ] No [ ] N/A [ ]
(23j)*	When transmitting untagged frames and the canonical_format_indicator parameter indicates that the mac_service_data_unit may contain embedded MAC Addresses in a format inappropriate to the destination MAC method, which of the following procedures is adopted by the Bridge:		7.1, 7.1.2.2	
(23j.1)	Convert any embedded MAC Addresses in the mac_service_data_unit to the format appropriate to the destination MAC method.	O.7		Yes [ ] No [ ]
(23j.2)	Discard the frame without transmission on that Port.	O.7		Yes [ ] No [ ]
(23k)	Does the Bridge perform frame translations, where necessary, in accordance with the procedures described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390?	TB:M	7.1, 7.1.2.2	Yes [ ] No [ ] N/A [ ]

Predicates:
TB = True if the Bridge supports translational Bridging; i.e., the Bridge supports 802.3/Ethernet MAC methods on one or more Ports and Token Ring/FDDI MAC methods on one or more Ports.

A.6 Relay and filtering of frames

Item	Feature	Status	References	Support
(2f)	Are received frames with MAC method errors discarded?	M	{D}6.4, 8.5	Yes [ ]
(2g)	Are correctly received frames submitted to the Learning Process?	M	8.5	Yes [ ]
(2h)	Are user data frames the only type of frame relayed?	M	8.5	Yes [ ]
(2i)	Are request with no response frames the only frames relayed?	M	8.5	Yes [ ]
(2j)	Are all frames addressed to the Bridge Protocol Entity submitted to it?	M	8.5	Yes [ ]
(2k)	Are user data frames the only type of frame transmitted?	M	8.9	Yes [ ]
(2l)	Are request with no response frames the only frames transmitted?	M	8.9	Yes [ ]
(2m)	Are relayed frames queued for transmission only under the conditions in 8.7.3?	M	8.7.3, {D}8.4	Yes [ ]

A.6 Relay and filtering of frames *(Continued)*

Item	Feature	Status	References	Support
(2n)	Is the order of relayed frames preserved in accordance with the requirements of the forwarding process?	M	8.7.3, 8.1.1	Yes [ ]
(2o)	Is a relayed frame submitted to a MAC Entity for transmission only once?	M	8.7.4, {D}6.3.4	Yes [ ]
(2p)	Is a maximum bridge transit delay enforced for relayed frames?	M	8.7.3	Yes [ ]
(2q)	Are queued frames discarded if a Port leaves the Forwarding State?	M	8.7.3	Yes [ ]
(2r)	Is the user priority of relayed frames preserved where possible?	M	{D}6.4	Yes [ ]
(2s)	Is the user priority set to the Default User Priority for the reception Port otherwise?	M	{D}6.4	Yes [ ]
(2t)	Is the user priority regenerated by means of the User Priority Regeneration Table?	M	8.5.1, Table 8-1	Yes [ ]
(2u)	Is mapping of Regenerated User Priority to Traffic Class performed by means of the Traffic Class Table?	M	8.7.3, Table 8-2	Yes [ ]
(2v)	Is the access priority derived from the Regenerated User Priority as defined by the values in Table 8-3 for each outbound MAC method supported by the Bridge?	M	8.7.5, Table 8-3	Yes [ ]
(2w)	Does the Bridge generate an M_UNITDATA.indication primitive on receipt of a valid frame transmitted by the Bridge Port's local MAC entity?	MS1:X	{D}6.5.4, ISO 9314-2	No [ ] N/A [ ]
(2x)	Is only Asynchronous service used?	MS1:M	ISO 9314-2, 8.1.4	Yes [ ] N/A [ ]
(2y)	On receiving a frame from an FDDI ring for forwarding, does the bridge set the C indicator?	MS1:O	{D}6.5.4, ISO 9314-2, 7.3.8	Yes [ ] No [ ] N/A [ ]
(2z)	On receiving a frame from an FDDI ring for forwarding, does the bridge leave the C indicator unaltered?	MS1:O	{D}6.5.4, ISO 9314-2, 7.3.8	Yes [ ] No [ ] N/A [ ]
	If item 4 is not supported, mark "N/A" and continue at item (4d).			N/A [ ]
(4a)*	Can the Default User Priority parameter for each Port be set to any value in the range 0 through 7?	4:O.6	{D}6.4	Yes [ ] No [ ]
(4b)*	Can the entries in the User Priority Regeneration Table for each Port be set to the full range of values shown in Table 8-1?	4:O.6	8.5.1, Table 8-1	Yes [ ] No [ ]
(4c)*	Can the entries in the Traffic Class Table for each Port be set to the full range of values shown in Table 8-2?	MS2:O	8.7.3, Table 8-2	Yes [ ] No [ ] N/A [ ]

A.6 Relay and filtering of frames *(Continued)*

Item	Feature	Status	References	Support
	If item 4 is supported, mark “N/A” and continue at item (4g)			N/A [ ]
(4d)	Does the Bridge support the recommended default value of the Default User Priority parameter for each Port?	¬ 4:M	{D}6.4	Yes [ ]
(4e)	Does the Bridge support the recommended default mappings between received user priority and Regenerated User Priority for each Port as defined in Table 8-1?	¬ 4:M	8.5.1, Table 8-1	Yes [ ]
(4f)	Does the Bridge support the recommended default user_priority to traffic class mappings shown in Table 8-2 for each Port?	MS3:M	8.7.3, Table 8-2	Yes [ ] N/A [ ]
(4g)	Is the Bridge able to use any values other than those shown in Table 8-3 when determining the access priority for the MAC methods shown?	X	8.7.5, Table 8-3	No [ ]

Predicates:
MS1 = 1a.4 AND NOT (1a.1 OR 1a.2 OR 1a.3 OR 1a.5 OR 1a.6 OR 1a.7 OR 1a.8)
MS2 = 2e AND 4
MS3 = 2e AND NOT 4

A.7 Maintenance of filtering entries in the Filtering Database

Item	Feature	Status	References	Support
(5a)	Are Dynamic Filtering Entries created and updated if and only if the Port State permits?	M	8.10, 8.11.3, {D}8.4	Yes [ ]
(5b)	Are Dynamic Filtering Entries created on receipt of frames with a group source address?	X	8.10, 8.11.3	No [ ]
(5c)	Does the Filtering Database support Static Filtering Entries?	M	8.11.1	Yes [ ]
(5d)	Can a Dynamic Filtering Entry be created that conflicts with an existing Static Filtering Entry?	X	8.10, 8.11, 8.11.1, 8.11.3	No [ ]
(5e)	Does the Filtering Database support Dynamic Filtering Entries?	M	8.11.3	Yes [ ]
(5f)	Does the creation of a Static Filtering Entry remove any conflicting information in a Dynamic Filtering Entry for the same address?	M	8.11.1, 8.11.3	Yes [ ]
(5g)	Does each Static Filtering Entry specify a MAC Address specification and a Port Map?	M	8.11.1	Yes [ ]
(5h)	Are Dynamic Filtering Entries removed from the Filtering Database if not updated for the Ageing Time period?	M	8.11.3	Yes [ ]
(5i)	Does each Dynamic Filtering Entry specify a MAC Address specification and a Port Map?	M	8.11.3	Yes [ ]

A.7 Maintenance of filtering entries in the Filtering Database *(Continued)*

Item	Feature	Status	References	Support
(5j)	Is the Filtering Database initialized with the entries contained in the Permanent Database?	M	8.11.10	Yes [ ]
	If item (2c) is not supported, mark N/A and continue at item (6a).			N/A [ ]
(5k)	Does each Group Registration Entry specify a MAC Address specification and a Port Map?	2c:M	8.11.4	Yes [ ]
(5l)	Can the MAC Address specification in Group Registration Entries represent All Groups, All Unregistered Groups, or a specific group MAC Address?	2c:M	8.11.4	Yes [ ]
(5m)	Are Group Registration Entries created, updated and removed from the Filtering Data-base in accordance with the specification of GMRP?	2c:M	8.11.4, {D}10	Yes [ ]
(5n)	Are Group Registration Entries created, updated and removed from the Filtering Data-base by any means other than via the operation of GMRP?	2c:X	8.11.4, {D}10	No [ ]
(6a)	State the Filtering Database Size.	M	8.11	____ entries
(6b)	State the Permanent Database Size.	M	8.11	____ entries
	If item (7c) is not supported, mark N/A and continue at item (8a).			N/A [ ]
(7d)	Can Static Filtering Entries be made for indi-vidual MAC Addresses?	**7c:**M	8.11.1	Yes [ ]
(7e)	Can Static Filtering Entries be made for group MAC Addresses?	**7c:**M	8.11.1	Yes [ ]
(7f)	Can a Static Filtering Entry be made for the broadcast MAC Address?	**7c:**M	8.11.1	Yes [ ]
(8a)	Can the Bridge be configured to use the default value of Ageing Time recommended in Table 8-4?	O	8.11.3, Table 8-4	Yes [ ] No [ ]
(8b)	Can the Bridge be configured to use any of the range of values of Ageing Time specified in Table 7-4?	O	8.11.3, Table 8-4	Yes [ ] No [ ]

A.8 Addressing

Item	Feature	Status	References	Support
(10a)	Does each Port have a separate MAC Address?	M	8.14.2	Yes [ ]
(10b)	Are all BPDUs transmitted to the same group address?	M	8.14.3, {D}8.2	Yes [ ]
	If item (9a) is not supported, mark N/A and continue at item (10d1).			N/A [ ]

A.8 Addressing *(Continued)*

Item	Feature	Status	References	Support
(10c)	Are all BPDUs transmitted to the Bridge Protocol Group Address when Universal Addresses are used?	**9a:**M	8.14.3, {D}8.2	Yes [ ]
(10d)	Is the source address of BPDUs the address of the transmitting Port?	**9a:**M	8.14.3	Yes [ ]
(10d1)	Is the LLC address of BPDUs the standard LLC address identified for the Spanning Tree Protocol?	M	8.14.3, Table 8-9	Yes [ ]
(10e)	Is the Bridge Address a Universal Address?	**M**	8.14.5, {D}8.2	Yes [ ] N/A [ ]
(10f)	Are frames addressed to any of the Reserved Addresses relayed by the Bridge?	X	8.14.6	No [ ]
	If item (13) is not supported, mark N/A and continue at item (11c).			N/A [ ]
(11a)	Is Bridge Management accessible through each Port using the MAC Address of the Port and the LSAP assigned?	**13:**O	8.14.4	Yes [ ] No [ ]
(11b)	Is Bridge Management accessible through all Ports using the All LANs Bridge Management Group Address?	13:O	8.14.4	Yes [ ] No [ ]
(11c)	Is the Bridge Address the Address of Port 1?	**9a:**O	8.14.5	Yes [ ] No [ ] N/A [ ]
(11d)*	Are Group Addresses additional to the Reserved Addresses pre-configured in the Permanent Database?	O	8.14.6	Yes [ ] No [ ]
	If item (11d) is not supported, mark N/A and continue at item (12a).			N/A [ ]
(11e)	Can the additional pre-configured entries in the Filtering Database be deleted?	**11d:**O	8.14.6	Yes [ ] No [ ]
(12a)	Can a group MAC Address be assigned to identify the Bridge Protocol Entity?	**9b:**M	{D}8.2	Yes [ ] N/A [ ]
(12c)	Does each Port of the Bridge have a distinct identifier?	M	{D}8.2, {D}8.5.5.1	Yes [ ]

A.9 Spanning Tree Algorithm

Item	Feature	Status	References	Support
(13a)	Are all the following Bridge Parameters maintained?	M	{D}8.5.3	Yes [ ]
	Designated Root		{D}8.5.3.1	
	Root Cost		{D}8.5.3.2	
	Root Port		{D}8.5.3.3	
	Max Age		{D}8.5.3.4	

A.9 Spanning Tree Algorithm *(Continued)*

Item	Feature	Status	References	Support
	Hello Time		{D}8.5.3.5	
	Forward Delay		{D}8.5.3.6	
	Bridge Identifier		{D}8.5.3.7	
	Bridge Max Age		{D}8.5.3.8	
	Bridge Hello Time		{D}8.5.3.9	
	Bridge Forward Delay		{D}8.5.3.10	
	Topology Change Detected		{D}8.5.3.11	
	Topology Change		{D}8.5.3.12	
	Topology Change Time		{D}8.5.3.13	
	Hold Time		{D}8.5.3.14	
(13b)	Are all the following Bridge Timers maintained?	M	{D}8.5.4	Yes [ ]
	Hello Timer		{D}8.5.4.1	
	Topology Change Notification Timer		{D}8.5.4.2	
	Topology Change Timer		{D}8.5.4.3	
(13c)	Are all the following Port Parameters maintained for each Port?	M	{D}8.5.5	Yes [ ]
	Port Identifier		{D}8.5.5.1	
	State		{D}8.5.5.2, {D}8.4	
	Path Cost		{D}8.5.5.3	
	Designated Root		{D}8.5.5.4	
	Designated Cost		{D}8.5.5.5	
	Designated Bridge		{D}8.5.5.6	
	Designated Port		{D}8.5.5.7	
	Topology Change Acknowledge		{D}8.5.5.8	
	Configuration Pending		{D}8.5.5.9	
	Change Detection Enabled		{D}8.5.5.10	
(13d)	Are all the following Timers maintained for each Port?	M	{D}8.5.6	Yes [ ]
	Message Age Timer		{D}8.5.6.1	
	Forward Delay Timer		{D}8.5.6.2	
	Hold Timer		{D}8.5.6.3	

A.9 Spanning Tree Algorithm *(Continued)*

Item	Feature	Status	References	Support
(13e)	Are Protocol Parameters and Timers maintained, and BPDUs transmitted, as required on each of the following events?	M	{D}8.7, {D}8.9, {D}8.5.3, {D}8.5.4, {D}8.5.5, {D}8.5.6	Yes [ ]
	Received Configuration BPDU		{D}8.7.1	
	Received Topology Change Notification BPDU		{D}8.7.2	
	Hello Timer Expiry		{D}8.7.3	
	Message Age Timer Expiry		{D}8.7.4	
	Forward Delay Timer Expiry		{D}8.7.5	
	Topology Change Notification Timer Expiry		{D}8.7.6	
	Topology Change Timer Expiry		{D}8.7.7	
	Hold Timer Expiry		{D}8.7.8	
(13f)	Do the following operations modify Protocol Parameters and Timers, and transmit BPDUs as required?	M	{D}8.8, {D}8.9, {D}8.5.3, {D}8.5.4, {D}8.5.5, {D}8.5.6	Yes [ ]
	Initialization		{D}8.8.1	
	Enable Port		{D}8.8.2	
	Disable Port		{D}8.8.3	
	Set Bridge Priority		{D}8.8.4	
	Set Port Priority		{D}8.8.5	
	Set Path Cost		{D}8.8.6	
(13g)	Does the implementation support the ability to set the value of the Change Detection Enabled parameter to Disabled?	O	{D}8.5.5.10	Yes [ ] No [ ]
(14a)	Does the Bridge underestimate the increment to the Message Age parameter in transmitted BPDUs?	X	{D}8.10.1	No [ ]
(14b)	Does the Bridge underestimate Forward Delay?	X	{D}8.10.1	No [ ]
(14c)	Does the Bridge overestimate the Hello Time interval?	X	{D}8.10.1	No [ ]
(15a)	Does the Bridge use the specified value for Hold Time?	M	{D}8.10.2, {D}Table 8-3	Yes [ ]
	If item (16) is not supported, mark N/A and continue at (17a).			N/A [ ]
(16a)	Can the relative priority of the Bridge be set?	**16:**M	{D}8.2, {D}8.5.3.7, {D}8.8.4	Yes [ ]

A.9 Spanning Tree Algorithm *(Continued)*

Item	Feature	Status	References	Support
(16b)	Can the relative priority of the Ports be set?	**16:**M	{D}8.2, {D}8.5.5.1, {D}8.8.5	Yes [ ]
(16c)	Can the path cost for each Port be set?	**16:**M	{D}8.2, {D}8.5.5.3, {D}8.8.6	Yes [ ]
	If item (17) is not supported, mark N/A and continue at (18a).			N/A [ ]
(17a)	Can Bridge Max Age be set to any of the range of values specified?	**17:**M	{D}8.10.2, {D}8.5.3.8, {D}Table 8-3	Yes [ ]
(17b)	Can Bridge Hello Time be set to any of the range of values specified?	**17:**M	{D}8.10.2, {D}8.5.3.9, {D}Table 8-3	Yes [ ]
(17c)	Can Bridge Forward Delay be set to any of the range of values specified?	**17:**M	{D}8.10.2, {D}8.5.3.10, {D}Table 8-3	Yes [ ]
(18a)	Do all BPDUs contain an integral number of octets?	M	{D}9.1.1	Yes [ ]
(18b)	Are all the following BPDU parameter types encoded as specified?	M	{D}9.1.1, {D}9.2	Yes [ ]
	Protocol Identifiers		{D}9.2.1	
	Protocol Version Identifiers		{D}9.2.2	
	BPDU Types		{D}9.2.3	
	Flags		{D}9.2.4	
	Bridge Identifiers		{D}9.2.5	
	Root Path Cost		{D}9.2.6	
	Port Identifiers		{D}9.2.7	
	Timer Values		{D}9.2.8	
(18c)	Do Configuration BPDUs have the format and parameters specified?	M	{D}9.3.1	Yes [ ]
(18d)	Do Topology Change Notification BPDUs have the format and parameters specified?	M	{D}9.3.2	Yes [ ]
(18e)	Are received BPDUs validated as specified?	M	{D}9.3.3	Yes [ ]

A.10 Bridge Management

Item	Feature	Status	References	Support
	If item (19) is not supported, mark N/A and continue at (20c).			N/A [ ]
(19a)	Discover Bridge	**19:**M	12.4.1.1	Yes [ ]
(19b)	Read Bridge	**19:**M	12.4.1.2	Yes [ ]
(19c)	Set Bridge Name	**19:**M	12.4.1.3	Yes [ ]
(19d)	Reset Bridge	**19:**M	12.4.1.4	Yes [ ]
(19e)	Read Port	**19:**M	12.4.2.1	Yes [ ]
(19f)	Set Port Name	**19:**M	12.4.2.2	Yes [ ]
(19g)	Read Forwarding Port Counters	**19:**M	12.6.1.1	Yes [ ]
(19g.1)	Are the Forwarding Port Counters maintained per VLAN?	19:O		Yes [ ] No [ ]
(19g.2)	Does the implementation support the Discard on Error Details parameter?	19:O		Yes [ ] No[ ]
(19h)	Read Filtering Database	**19:**M	12.7.1.1	Yes [ ]
(19i)	Set Filtering Database Ageing Time	**19:**M	12.7.1.2	Yes [ ]
(19j)	Read Permanent Database	**19:**M	12.7.6.1	Yes [ ]
(19k)	Create Filtering Entry	**19:**M	12.7.7.1	Yes [ ]
(19l)	Delete Filtering Entry	**19:**M	12.7.7.2	Yes [ ]
(19m)	Read Filtering Entry	**19:**M	12.7.7.3	Yes [ ]
(19n)	Read Filtering Entry Range	**19:**M	12.7.7.4	Yes [ ]
(19o)	Read Bridge Protocol Parameters	**19:**M	12.8.1.1	Yes [ ]
(19p)	Set Bridge Protocol Parameters	**19:**M	12.8.1.2	Yes [ ]
(19q)	Read Port Parameters	**19:**M	12.8.2.1	Yes [ ]
(19r)	Force Port State	**19:**M	12.8.2.2	Yes [ ]
(19s)	Set Port Parameters	**19:**M	12.8.2.3	Yes [ ]
(19t)	Read Port Default User Priority	MS4:M	12.6.2.1	Yes [ ] N/A [ ]
(19u)	Set Port Default User Priority	MS4:M	12.6.2.2	Yes [ ] N/A [ ]
(19v)	Read Port User Priority Regeneration Table	MS5:M	12.6.2.3	Yes [ ] N/A [ ]
(19w)	Set Port User Priority Regeneration Table	MS5:M	12.6.2.3	Yes [ ] N/A [ ]
(19x)	Read Port Traffic Class Table	MS7:M	12.6.3.1	Yes [ ] N/A [ ]
(19y)	Set Port Traffic Class Table	MS7:M	12.6.3.1	Yes [ ] N/A [ ]
(19z)	Read Outbound Access Priority Table	MS6:M	12.6.2.5	Yes [ ] N/A [ ]
(19aa)	Read GARP Timers	MS8:M	12.9.1.1	Yes [ ] N/A [ ]
(19ab)	Set GARP Timers	MS8:M	12.9.1.2	Yes [ ] N/A [ ]
(19ac)	Read GARP Protocol Controls	MS8:M	12.9.2.1	Yes [ ] N/A [ ]

A.10 Bridge Management *(Continued)*

Item	Feature	Status	References	Support
(19ad)	Set GARP Protocol Controls	MS8:M	12.9.2.2	Yes [ ] N/A [ ]
(19ae)	Read GARP State	MS8:M	12.9.3.1	Yes [ ] N/A [ ]
(19af)	Read Bridge VLAN Configuration	19:M	12.10.1.1	Yes [ ] N/A [ ]
(19ah)	Configure PVID values	19:M	12.10.1.2	Yes [ ] N/A [ ]
(19ai)	Configure Acceptable Frame Types parameter	23a.2:M	12.10.1.3	Yes [ ] N/A [ ]
(19aj)	Configure Enable Ingress Filtering parameters	23g:M	12.10.1.4	Yes [ ] N/A [ ]
(19ak)	Reset Bridge VLAN Bridge.	19:M	12.10.1.5	Yes [ ] N/A [ ]
(19al)	Notify VLAN Registration Failure	19:M	12.10.1.6	Yes [ ] N/A [ ]
(19am)	Read VLAN Configuration	19:M	12.10.2.1	Yes [ ] N/A [ ]
(19an)	Create VLAN Configuration	19:M	12.10.2.2	Yes [ ] N/A [ ]
(19ao)	Delete VLAN Configuration	19:M	12.10.2.3	Yes [ ] N/A [ ]
	If Item (23e.6) is not supported, mark N/A and continue at Item (19at).			N/A
(19ap)	Read VLAN Learning Constraints	23e.6:M	12.10.3.1	Yes [ ]
(19aq)	Read VLAN Learning Constraints for VID	23e.6:M	12.10.3.2	Yes [ ]
(19aq)	Set VLAN Learning Constraint	23e.6:M	12.10.3.3	Yes [ ]
(19ar)	Delete VLAN Learning Constraint	23e.6:M	12.10.3.4	Yes [ ]
(19as)	Notify Learning Constraint Violation	23e.6:M	12.10.3.10	Yes [ ]
	If Item (23e.8) is not supported, mark N/A and continue at Item (20c).			N/A
(19at)	Read VID to FID allocations	23e.8:M	12.10.3.5	Yes [ ]
	Read FID allocation for VID	23e.8:M	12.10.3.6	Yes [ ]
	Read VIDs allocated to FID	23e.8:M	12.10.3.7	Yes [ ]
	Set VID to FID allocation	23e.8:M	12.10.3.8	Yes [ ]
	Delete VID to FID allocation	23e.8:M	12.10.3.9	Yes [ ]
	If item (20a) is not supported, mark N/A and continue at (20e).	23e.8:M		N/A [ ]
(20c)	What Management Protocol standard(s) or specification(s) are supported?	**20a:**M	{D}5	
(20d)	What standard(s) or specifications for Managed Objects and Encodings are supported?	**20a:**M	{D}5	

A.10 Bridge Management *(Continued)*

Item	Feature	Status	References	Support
	If item (20b) is not supported, mark N/A and continue at A.11.			N/A [ ]
(20e)	What specification of the local management interface is supported?	**20b:**M	{D}5	

Predicates:
MS4=19 AND 4a
MS5=19 AND 4b
MS6=19 AND 4
MS7=19 AND 4c
MS8=19 AND 2b

A.11 Performance

Item	Feature	Status	References	Support
(21a)	Specify a Guaranteed Port Filtering Rate, and the associated measurement interval *TF*, for each Bridge Port in the format specified below.	M	{D}16.1	
(21b)	Specify a Guaranteed Bridge Relaying Rate, and the associated measurement interval *TR*, in the format specified below. Supplementary information shall clearly identify the Ports.	M	{D}16.2	

Guaranteed Bridge Relaying Rate	TR
_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

Port number(s) or other identification	Guaranteed port filtering rate (specify for all ports)	T_F (specify for all ports)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

Port number(s) or other identification	Guaranteed port filtering rate (specify for all ports)	T_F (specify for all ports)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)
	_ _ _ _ _ _ _ _ _ _ _ frames per second	_ _ _ _ _ _ second(s)

A.12 GARP and GMRP

Item	Feature	Status	References	Support
	If Item 2b is not supported, mark N/A and continue at item (22i).			N/A [ ]
(22a)	Is the GMRP Application address used as the destination MAC Address in all GMRP protocol exchanges?	2b:M	{D}10.4.1, {D}Table 12-1	Yes [ ]
(22b)	Are GMRP protocol exchanges achieved by means of LLC Type 1 procedures, using the LLC address for Spanning Tree protocol?	2b:M	{D}12.4, {D}12.5, {D}Table 7-8	Yes [ ]
(22c)	Are GMRP protocol exchanges achieved using the GARP PDU formats, and the definition of the attribute type and value encodings defined for GMRP?	2b:M	10, {D}10.3.1, {D}12.4, {D}12.5, {D}12.11	Yes [ ]
(22d)	Does the implementation support the operation of the Applicant, Registrar, and Leave All state machines?	2b:M	{D}12.8	Yes [ ]
(22e)	Does the Bridge propagate registration GMRP information only on Ports that are part of the active topology of the GIP Context for the VLAN on which the registration was received?	2b:M	10, {D}12.3.3, {D}12.3.4	Yes [ ]
(22f)	Are GARP PDUs received on Ports that are in the Forwarding State forwarded, filtered or discarded in accordance with the requirements for handling GARP Application addresses?	2b:M	{D}7.12.3, {D}12.5	Yes [ ]
(22g)	Does the GMRP application operate as defined in Clause 10 of ISO/IEC 15802-3, as modified by Clause 10 of this standard?	2b:M	10, {D}10, {D}10.3	Yes [ ]
(22h)	Are received GARP PDUs that are not well formed for any GARP Applications supported, discarded?	2b:M	10, {D}10.3.1, {D}12.4, {D}12.5, {D}12.10, {D}12.11	Yes [ ]
(22i)	Are all GARP PDUs that are (a) Received on Ports that are in the Forwarding State, and are (b) Destined for GARP applications that the Bridge does not support, forwarded on all other Ports that are in Forwarding?	M	8.14.3, {D}12.5	Yes [ ]

A.12 GARP and GMRP *(Continued)*

Item	Feature	Status	References	Support
(22j)	Are any GARP PDUs that are (a) Received on any Port, and (b) Destined for GARP applications that the Bridge does not support, submitted to any GARP Participants?	X	8.14.3, {D}12.5	No [ ]
(22k)	Are any GARP PDUs that are (a) Received on any Ports that are not in the Forwarding State, and are (b) Destined for GARP applications that the Bridge does not support, forwarded on any other Ports of the Bridge?	X	8.14.3, {D}12.5	No [ ]
(22l)	Are any GARP PDUs that are (a) Received on any Ports that are in the Forwarding State, and are (b) Destined for GARP applications that the Bridge supports, forwarded on any other Ports of the Bridge?	X	8.14.3, {D}12.5	No [ ]
(22m)	Are all GARP PDUs that are: (a) Received on any Port, and (b) Destined for GARP applications that the Bridge supports, submitted to the appropriate GARP Participants?	M	8.14.3, {D}12.5	Yes [ ]

A.13 VLAN support

Item	Feature	Status	References	Support
	Ingress rules			
(24a)	Can the PVID for any Port be assigned the value of the null VLAN ID?	X	8.4.4, Table 9-2	No [ ]
(24b)	Are frames discarded (or not discarded) in accordance with the settings of the Acceptable Frame Types parameters?	M	8.6	Yes [ ]
(24c)	Are all frames received classified as belonging to exactly one VLAN, as defined in the ingress rules?	M	8.6	Yes [ ]
(24d)	Is Ingress Filtering performed in accordance with the value of the Enable Ingress Filtering parameter?	M	8.6	Yes [ ]
(24e)	Are all frames that are not discarded as a result of the application of the ingress rules submitted to the Forwarding Process and to the Learning Process?	M	8.6	Yes [ ]
	Egress rules			
(25a)	Are frames discarded if the transmission Port is not present in the Member set for the frame's VID?	M	8.8, 8.11.9	Yes [ ]

A.13 VLAN support *(Continued)*

Item	Feature	Status	References	Support
(25b)	Are frames discarded if the value of the include_tag parameter is False, and the Bridge does not support the ability to translate embedded MAC Address information from the format indicated by the canonical_format_indicator parameter to the format appropriate to the media type on which the data request will be carried?	23j.2:M	8.8	Yes [ ] N/A [ ]
(25c)	Are frames transmitted as VLAN-tagged frames or as untagged frames in accordance with the value of the untagged set for the frame's VID?	M	8.8	Yes [ ]
	Filtering Database			
(26a)	Does the implementation support Static VLAN Registration Entries as defined in 8.11.2?	M	8.11.2	Yes [ ]
(26b)	Does the implementation support the creation of a separate Static VLAN Registration Entry with a distinct Port Map for each VLAN from which frames are received by the Forwarding Process?	O	8.11.2	Yes [ ] No [ ]
(26c)	Does the implementation support Dynamic VLAN Registration Entries as defined in 8.11.5?	M	8.11.5	Yes [ ]
(26d)	Does the implementation support the creation of a separate Dynamic VLAN Registration Entry with a distinct Port Map for each VLAN from which frames are received by the Forwarding Process?	O	8.11.5	Yes [ ] No [ ]
(26e)	Does the implementation allocate VIDs to FIDs in accordance with the specification in 8.11.7?	M	8.11.7, 8.11.7.2	Yes [ ]
(26f)	Does the implementation correctly detect Learning Constraint violations?	M	8.11.7.3	Yes [ ]
(26g)	Is determination of the Member set and the untagged set for a given VLAN achieved as defined in 8.11.9?	M	8.11.9	Yes [ ]
	Tagged frames			
(27a)	Do VLAN-tagged frames transmitted by the Bridge conform to the format defined in Clause 9 for the MAC type on which they are transmitted?	M	9	Yes [ ]
(27b)	Are all BPDUs transmitted untagged?	M	8.14.7	Yes [ ]
	VLAN use of GMRP. If item (2b) is not supported, mark N/A and continue at item (29a).			N/A [ ]
(28a)	Does the implementation of GMRP recognize the use of VLAN Contexts for the transmission and reception of GMRP PDUs?	2b:M	10, 10.1, 10.2 10.3	Yes [ ]

A.13 VLAN support *(Continued)*

Item	Feature	Status	References	Support
(28b)	Does the implementation of GMRP support the creation of distinct GMRP Participants for each VLAN context?	2b:M	10.2	Yes [ ]
(28c)	Does the implementation support the identification of VLAN contexts in transmitted GMRP PDUs by means of VLAN-tagged or untagged frames, in accordance with the member set and untagged set for the VLAN Context concerned?	2b:M	10.3	Yes [ ]
(28d)	Are GMRP PDUs transmitted only on Ports that are part of the active topology for the VLAN Context concerned?	2b:M	10.1	Yes [ ]
	VLAN Topology Management			
(29a)	Does the implementation support the creation, updating and removal of Dynamic VLAN Registration Entries in the Filtering Database under the control of GVRP?	M	11	Yes [ ]
(29b)	Does the Permanent Database contain an entry for the Default VID that defines Registration Fixed on all Ports?	O	11.2.1.3	Yes [ ] No [ ]
(29c)	Is the GVRP Application address used as the destination MAC Address in all GVRP protocol exchanges?	M	11, Table 11-1	Yes [ ]
(29d)	Are GVRP protocol exchanges achieved by means of LLC Type 1 procedures, using the LLC address for Spanning Tree protocol?	M	11, {D}12.4, {D}12.5, {D}Table 7-8	Yes [ ]
(29e)	Are GVRP protocol exchanges achieved using the GARP PDU formats, and the definition of the attribute type and value encodings defined for GVRP?	M	11, 11.2.3.1, {D}12.4, {D}12.5, {D}12.11	Yes [ ]
(29f)	Does the implementation support the operation of the Applicant, Registrar, and Leave All state machines?	M	{D}12.8	Yes [ ]
(29g)	Does the Bridge propagate registration GVRP information only on Ports that are part of the active topology of the base Spanning Tree Context?	M	11, {D}12.3.3, {D}12.3.4	Yes [ ]
(29h)	Does the GVRP application operate as defined in Clause 11?	M	11	Yes [ ]

Annex B

(informative)

Shared and Independent VLAN Learning

This standard provides for a variety of approaches to the implementation of VLAN Bridges from the point of view of the way that individual MAC Addresses are learned, and how that learned information is used in subsequent forwarding/filtering decisions. There are two mechanisms that are used as a basis for these variants:

a) Making use of address information learned across a number of VLANs in order to make learning decisions relative to any one of those VLANs. This is referred to as *Shared VLAN Learning* (SVL, 3.9);
b) Making use of address information learned in one VLAN only in order to make learning decisions relative to that VLAN, and ensuring that it is not used in learning decisions relative to any other VLAN. This is referred to as *Independent VLAN Learning* (IVL, 3.5).

These mechanisms lead to the SVL/IVL model for how a VLAN Bridge implements learning and filtering for MAC Addresses. Using the terminology of 8.11.7, an SVL/IVL Bridge supports multiple FIDs (which effectively equates to supporting multiple Filtering Databases), and multiple VLANs can use each FID. By varying the number of FIDs supported, and the number of VLANs that can share each FID, the following simplifications of the SVL/IVL model can be created:

c) *Shared VLAN Learning (SVL) only.* The implementation supports only one FID, so all VLANs share the same learned MAC Address information, regardless of which VLAN the information was learned in;
d) *Independent VLAN Learning (IVL) only.* Multiple FIDs are supported, but each FID can support only one VID, so each VLAN makes use only of MAC Address information learned within that VLAN.

All three approaches are permitted by this standard, and each has advantages in particular circumstances. The remainder of this annex discusses

e) The requirements for Independent VLAN Learning, Shared VLAN Learning, or both;
f) How Bridges are made aware of the requirement for particular VLANs to be “shared” or “independent”;
g) How Bridges based on one of these models can interoperate with Bridges based on a different model, in the same Bridged LAN.

B.1 Requirements for Shared and Independent Learning

Under most circumstances, the SVL and IVL approaches work equally well, and Bridges adopting either approach can be freely intermixed within a Bridged LAN. There are, however, a small number of configuration cases where, in order to prevent undue flooding of unicast frames, and in some cases, to make communication between the affected end systems possible, it is necessary to make specific choices as to how Bridges that adopt these different learning models are deployed in a Bridged LAN. The following subclauses give examples of some of these configurations, and also provide a generic statement of the requirements that must be met in order for each learning model to be successfully deployed.

B.1.1 Connecting independent VLANs

Figure B-1 illustrates how a device that connects two VLANs together, and which therefore itself shares learning between those VLANs, creates a need for those VLANs to be independent in other Bridges.

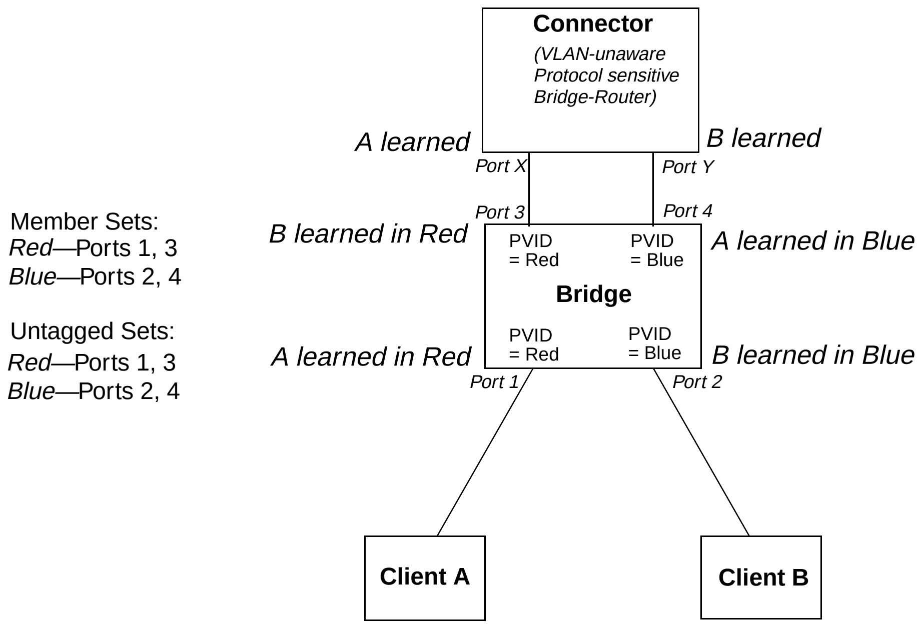


Figure B-1—Connecting independent VLANs—1

Clients A and B are connected via the protocol sensitive Bridge-Router (Connector), with an intervening VLAN-aware Bridge. The fact that all the Ports of the Bridge carry untagged traffic neatly conceals the fact that the Connector has the effect of connecting VLANs Red and Blue together with regard to bridged traffic. The Connector itself learns A and B in the same database, as it has no knowledge of VLANs Red and Blue. This prevents any traffic transmitted on the Red VLAN (Port X of the Connector) that is destined for A, from being bridged to Port Y and transmitted on the Blue VLAN.

The VLAN-aware Bridge must keep its learning separate for Red and Blue; otherwise, the addresses of the two clients would be alternately learned on diagonally opposite Ports as, for example, traffic sourced by A reenters the Bridge on Port 4 having previously been seen on Port 1.

NOTE—This example assumes that Spanning Tree is disabled in the Connector, so that the VLAN-aware Bridge does not attempt to suppress the loop that apparently exists if VLANs are not taken into account.

A simpler example can be constructed, with a single Port connecting the Connector and the VLAN-aware Bridge, if the Connector is itself VLAN-aware and transmits and receives only VLAN-tagged traffic. In this case, the Connector would allocate a single FID for use by Red and Blue. This is shown in Figure B-2.

B.1.2 Duplicate MAC Addresses

The simplest example of a need for Independent VLAN Learning occurs where two (or more) distinct devices in different parts of the network reuse the same individual MAC Address, or where a single device is connected to multiple LAN segments, and all of its LAN interfaces use the same individual MAC Address. This is shown in Figure B-3.

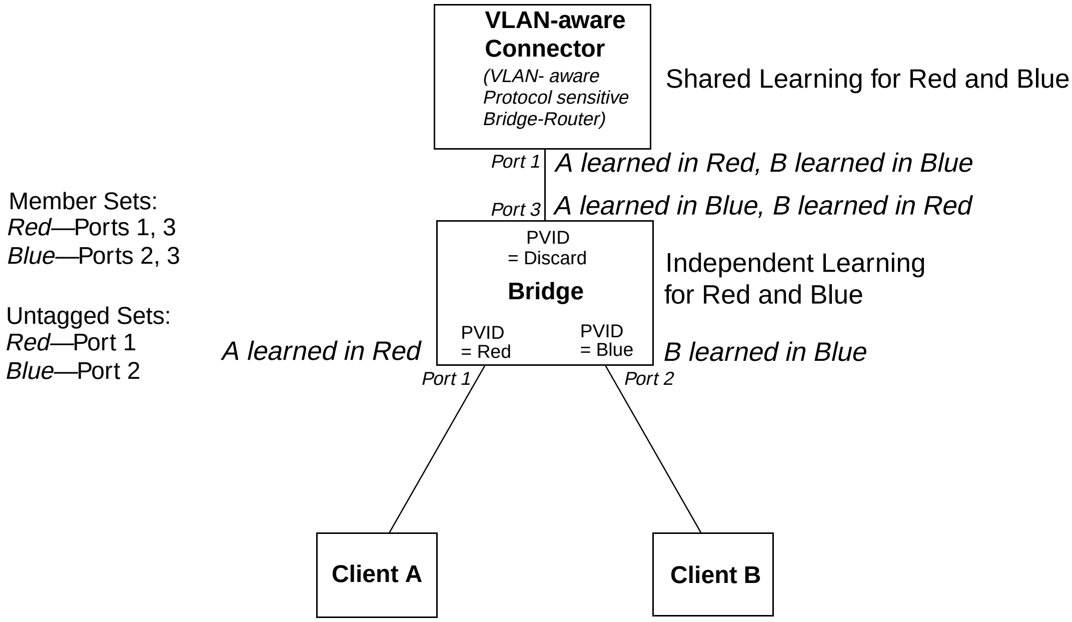


Figure B-2—Connecting independent VLANs—2

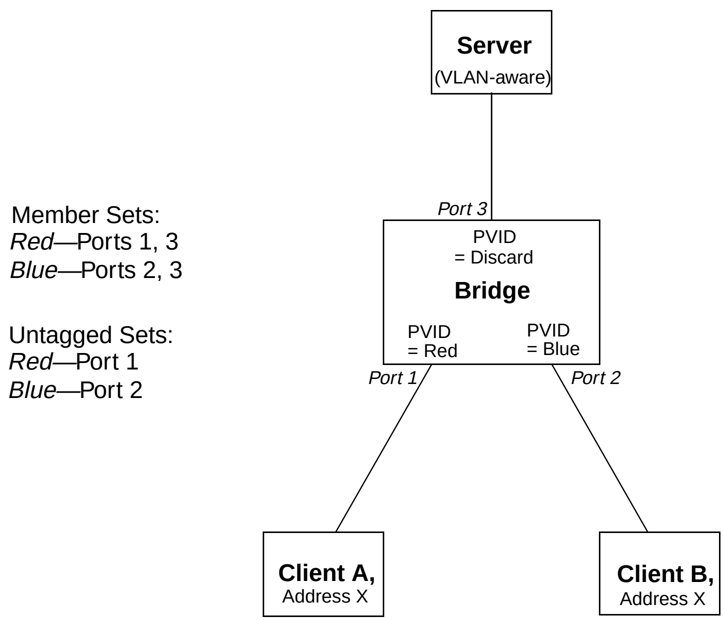


Figure B-3—Duplicate MAC Addresses

The example shows two clients with access to the same server; both clients are using the same individual MAC Address, X. If the Bridge shares learning between VLAN Red (which serves Client A) and VLAN Blue (which serves Client B), i.e., the Bridge uses the same FID for both VLANs, then Address X will appear to move between Ports 1 and 2 of the Bridge, depending upon which client has most recently transmitted a frame. Communication between these Clients and the server will therefore be seriously disrupted. Assignment of distinct FIDs for Red and Blue ensures that communication can take place correctly.

Hence, in order to construct this particular VLAN configuration, either an IVL Bridge or an SVL/IVL Bridge would be required.

B.1.3 Asymmetric VLANs

A primary example of the requirement for Shared VLAN Learning is found in “asymmetric” uses of VLANs. Under normal circumstances, a pair of devices communicating in a VLAN environment will both send and receive using the same VLAN; however, there are some circumstances in which it is convenient to make use of two distinct VLANs, one used for A to transmit to B: the other used for B to transmit to A. An example of such an application of VLANs is shown in Figure B-4. Note that:

a) In the example, the server and both clients are assumed to be VLAN-unaware devices, i.e., they transmit and receive untagged frames only;
b) The ingress classification rules assumed by the example are as defined in this standard, i.e., Port-based classification only;
c) The configuration shown can only be achieved by management configuration of appropriate values in Static VLAN Registration Entries (8.11.9) in order to configure the indicated member sets and untagged sets.

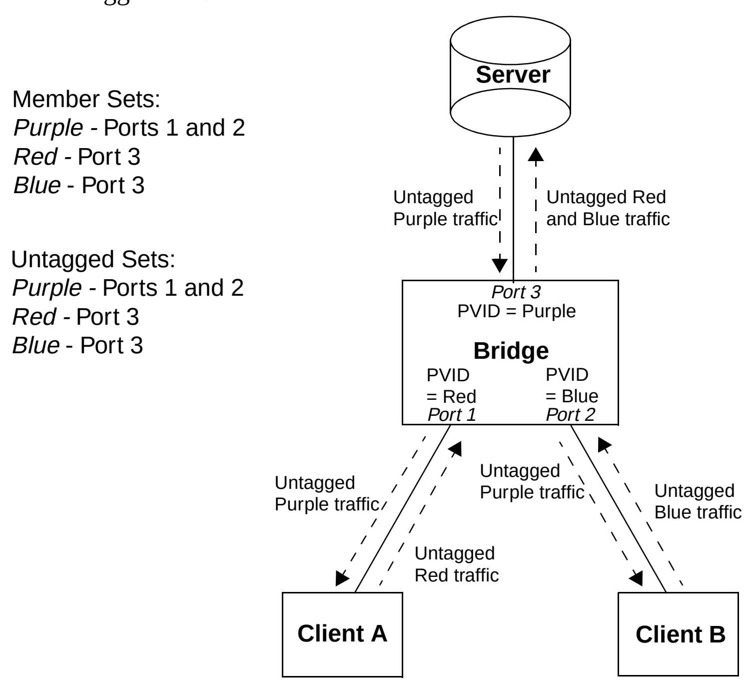


Figure B-4—Asymmetric VLAN use: “multi-netted server”

In the example, Port-based tagging and an asymmetric VLAN configuration is used in order to permit Clients A and B access to a common server, but to prohibit Clients A and B from talking to each other. Examples of where this type of configuration might be required are if the clients are on distinct IP subnets, or if there is some confidentiality-related need to segregate traffic between the clients.

Client A transmits to the server via Port 1, which will classify this traffic as belonging to VLAN Red; the Bridge therefore learns Client A’s MAC Address on Port 1 in VLAN Red. The Server transmits its responses to Client A via Port 3, which classifies the return traffic as belonging to VLAN Purple. If individual MAC Address learning is configured in the Bridge such that learning is independent between Red and Purple (Red

and Purple are allocated to distinct FIDs), then the Bridge will have no knowledge of A in VLAN Purple, and will therefore flood the server's responses to Client A on both Port 1 and Port 2. Conversely, if Red and Purple are defined to share the same FID, then the address information learned in Red will be available for use in forwarding the Purple traffic, and the responses to Client A are forwarded only through Port 1.

Similarly, there is a need in this configuration for Blue and Purple to share learning information; hence, in order for this configuration to achieve its objectives, the Red, Blue, and Purple VIDs must be allocated to the same FID in the Bridge.

Hence, in order to construct this particular VLAN configuration, either an SVL Bridge or an SVL/IVL Bridge would be required.

NOTE—The example has been deliberately simplified; in practical applications, the central Bridge would likely be replaced by a number of VLAN-aware Bridges, interconnected with links that would carry the traffic between clients and server as VLAN-tagged frames, with VLAN-tagging and untagging occurring only at the "edge" Ports of the Bridged LAN. An alternative approach to the one described here could also be achieved either by using a VLAN-aware server, or by use of more sophisticated Ingress classification rules.

B.1.4 Generic constraints on SVL and IVL use

This subclause describes the general constraints on the mapping of VLANs to FIDs, from the point of view of a given Bridge that learns from or forwards frames on a set of VLANs (the Bridge's "active set" of VLANs). If

a) The individual MAC Addresses associated with each point of attachment to the active set of VLANs are unique (i.e., the "Duplicate MAC Address problem" is not present), and
b) There is no Bridge or Bridge-like device that takes frames from one VLAN in the active set and subsequently transmits them on another VLAN in the active set, then

every VLAN in the active set may share the same FID; in other words, individual MAC Address information learned in any one VLAN may be used in forwarding decisions taken relative to any of the others, so the SVL approach can be used in that Bridge.

Further, if

c) Each bidirectional, individual MAC-Addressed, conversation between pairs of end stations makes use of the same VLAN (ID) in both directions, then

every VLAN in the active set may be allocated a distinct FID (with the possibility of a little extra flooding as learning of addresses in one VLAN does not contribute to forwarding decisions for that address in any other VLAN). Under these circumstances, rule b) may also be relaxed and restated as follows:

d) Frames on one VLAN in the active set may be received by (up to) one Bridge and transmitted on another VLAN in the active set, provided that there is no loop in such VLAN to VLAN forwarding, e.g., for a set of VLANs Red, Blue, Green,...etc., there is no logical loop in copying frames between VLANs, such as copying from Red to Green by one Bridge, Green to Blue by another, and Blue back to Red by a third.

So

e) If rules a), b), and c) are true, and d) is false, for all VLANs in the active set, then either an SVL or an IVL Bridge can be deployed;
f) If rules a) and b) are true, and c) and d) are false for all VLANs in the active set, then only an SVL Bridge can be deployed;

g) If rules a) or b) or d) are false, and c) is true for all VLANs in the active set, then only an IVL Bridge can be deployed.

The above conditions are all on the basis that they apply “for all VLANs in the active set.” Clearly, in more complex scenarios, some VLANs in the active set will have requirements that dictate SVL behavior on the part of a given Bridge, while others will have requirements that dictate IVL behavior. Under such circumstances, an SVL/IVL Bridge is required, allowing those VLANs that need to be shared to be mapped to a single FID, while those that need to be independent are mapped to distinct FIDs. Needless to say, wherever an SVL or IVL Bridge can be deployed, it can successfully be replaced by an appropriately configured SVL/IVL Bridge.

B.2 Configuring the Global VLAN Learning Constraints

Clause B.1 described the requirements that exist for the two approaches to learning in VLAN Bridges, closing with some generic rules for how to determine whether, for a given Bridge, SVL, IVL, or SVL/IVL can be successfully deployed. In VLAN Bridges, the set of requirements for Independent and/or Shared VLAN Learning is configured as a set of global VLAN Learning Constraint specifications, using the management tools defined in 12.10.3. Two types of VLAN Learning Constraint are defined in 8.11.7.2, which also defines how the set of constraints is used in order to derive a valid mapping of VIDs to FIDs.

The constraint specifications can be constructed on a modular basis. For example, the configuration shown in Figure B-4 has a requirement for Shared VLAN Learning between VLANs Red, Blue, and Purple. This could be expressed as follows:

{Red S Purple};
{Blue S Purple}

with {Red S Blue} being implied by the transitive nature of the S Constraint.

If we add a similar server access configuration in the same network that requires Red to share with Yellow and Orange, then this could be expressed as

{Red S Yellow};
{Orange S Yellow}

with {Red S Orange}, {Yellow S Blue}, {Yellow S Purple}, {Orange S Blue}, and {Orange S Purple} being implied by the transitive nature of the S Constraint.

Hence, Red, Blue, Purple, Yellow and Orange are all required to map to the same FID in order for the set of S Constraints (both explicit and implied) to be met. The constraints that express that requirement are built up from their constituent requirements; namely, for Red and Blue to share with Purple to meet one configuration need, and for Red to share with Yellow and Orange to meet another.

NOTE 1—The five VLANs in this example can be viewed as forming a Shared Set; i.e., a set of VLANs that have a mutual requirement to share learned information—all members of a Shared Set must map to the same FID. Any sequence of S Constraints defines one or more such Shared Sets. Any two Shared Sets can also map to the same FID as long as, for any pair of VLANs, one selected from each Shared Set, there are no I Constraints that require that pair of VLANs to learn independently. Hence, if there are only S Constraints defined, then all VLANs can be mapped to a single FID.

Similarly, the I Constraints can be added on a modular basis. Continuing from the above example, a Bridge-Router (Figure B-1) might be present in the Bridged LAN, which has the effect of connecting VLANs Indigo and Green together, thus creating a requirement for Indigo and Green to be independent. This could be expressed as

{Indigo I 1};
{Green I 1}

A separate independence requirement might be imposed by the fact that three stations, attached to Indigo, Vermilion, and Red VLANs, all make use of the same individual MAC Address. This could be expressed as:

{Indigo I 2};
{Vermilion I 2};
{Red I 2}

Hence, {Indigo, Vermilion, Red} have to be mutually independent (assigned to distinct FIDs), {Indigo, Green} have to be mutually independent, and {Red, Blue, Purple, Yellow, Orange} have to be shared.

The minimum number of FIDs required to satisfy this total constraint specification is three, e.g.:

FID A: Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Green, Vermilion

although an equally valid allocation for 3 FIDs is

FID A: Green, Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Vermilion

and an equally valid allocation, using the maximum number of FIDs that could be used for this set of constraints and VLANS is:

FID A: Red, Blue, Purple, Yellow, Orange
FID B: Indigo
FID C: Green
FID D: Vermilion

NOTE 2—It can clearly be seen from this example that it is possible to add further constraints that result in impossible VID to FID mappings; for example, if we were to add {Indigo S Red} or ({Yellow I 3}, {Blue I 3}), then the result is at least one pair of VLANs that have a requirement to both share the same FID and to use distinct FIDs at the same time. Such configurations are examples of Learning Constraint inconsistencies (8.11.7.3).

The assumption behind these constraint specifications is that they are applied globally, in the sense that all VLAN Bridges in a given Bridged LAN are configured with the same set of constraints. This is important, in order to ensure that each Bridge is in a position to determine whether or not, given its current active set of VLANs, it is capable of adopting a VID to FID mapping that will satisfy the specified constraints. If it cannot achieve such a mapping (for any of the reasons identified in 8.11.7.3), then it has detected a network misconfiguration that can only be resolved by management intervention. The managed object specification 12.10.3 provides a Notification for use in such circumstances, to alert a management station to the existence of the problem.

B.3 Interoperability

If the configuration of the Bridged LAN is such that it is not necessary to configure any VLAN Learning Constraints into the Bridges, i.e.:

a) There are no instances where two (or more) points of attachment to different LAN segments (and different VLANs) make use of the same individual MAC Address;
b) There are no instances where a Bridge receives frames on one VLAN and transmits them on another VLAN;
c) There is no asymmetric VLAN use, i.e., there is no pair of end stations for which bidirectional, unicast conversations make use of different VLANs for each direction of transmission,

then it is possible to freely intermix SVL, IVL, and SVL/IVL Bridges in that Bridged LAN, and they can all successfully interoperate.

If the configuration of the Bridged LAN requires one or more S Constraints (and no I Constraints) to be configured into the Bridges, then SVL and SVL/IVL Bridges can be used freely; however, IVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which an S Constraint (either explicit or implied) has been defined.

If the configuration of the Bridged LAN requires two or more I Constraints (and no S Constraints) to be configured into the Bridges, then IVL and SVL/IVL Bridges can be used freely; however, SVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which I Constraints with the same Independent Set Identifier have been defined.

If the configuration of the Bridged LAN requires both I Constraints and S Constraints to be configured into the Bridges, then SVL/IVL Bridges can be used freely; however,

d) SVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which I Constraints with the same Independent Set Identifier have been defined, and
e) IVL Bridges may only be used in locations where their active set of VLANs does not include any pair of VLANs for which an S Constraint (either explicit, or implied) has been defined.

Annex C

(informative)

MAC method dependent aspects of VLAN support

This annex examines the set of services, frame formats, and MAC methods involved in the provision of VLAN services across IEEE 802 LANs using different MAC methods, and the mapping/bridging functions necessary for that provision.

C.1 The variables

End station MAC Service users make use of MAC data transmission services that convey the following types of information:

a) Ethernet Type-encoded (E) and LLC-encoded (L) information (see 3.1 and 3.2);
b) Frames (either E or L) in which any MAC Addresses embedded in the MAC data are carried in Canonical (C) or Non-canonical (N) format;

NOTE 1—The terms *Canonical format* and *Non-canonical format* are described in Annex F.

c) Frames that carry source-routing information (R), or frames that are Bridged transparently (T).

Hence, there are potentially eight combinations of these variables, corresponding to eight distinct services, as follows:

d) E-C-T (Ethernet Type-encoded, Canonical, transparent),
e) E-C-R (Ethernet Type-encoded, Canonical, source-routed),
f) E-N-T (Ethernet Type-encoded, Non-canonical, transparent),
g) E-N-R (Ethernet Type-encoded, Non-canonical, source-routed),
h) L-C-T (LLC-encoded, Canonical, transparent),
i) L-C-R (LLC-encoded, Canonical, source-routed),
j) L-N-T (LLC-encoded, Non-canonical, transparent),
k) L-N-R (LLC-encoded, Non-canonical, source-routed),

These services are supported over two basic LAN types:

l) 802.3/Ethernet (C);
m) Token Ring/FDDI (R).

There are two VLAN environments involved:

n) Untagged frames (U);
o) Tagged frames (T).

This leads to a total of 32 potential frame/encapsulation formats to consider (8 X 2 X 2). There are 96 possible one-way heterogeneous bridging functions between these various LAN/VLAN environments; 48 symmetrical (2-way) functions.

The combination of services and environments is illustrated in Figure C-1; italics indicate services that have no untagged representation on the MAC method concerned.

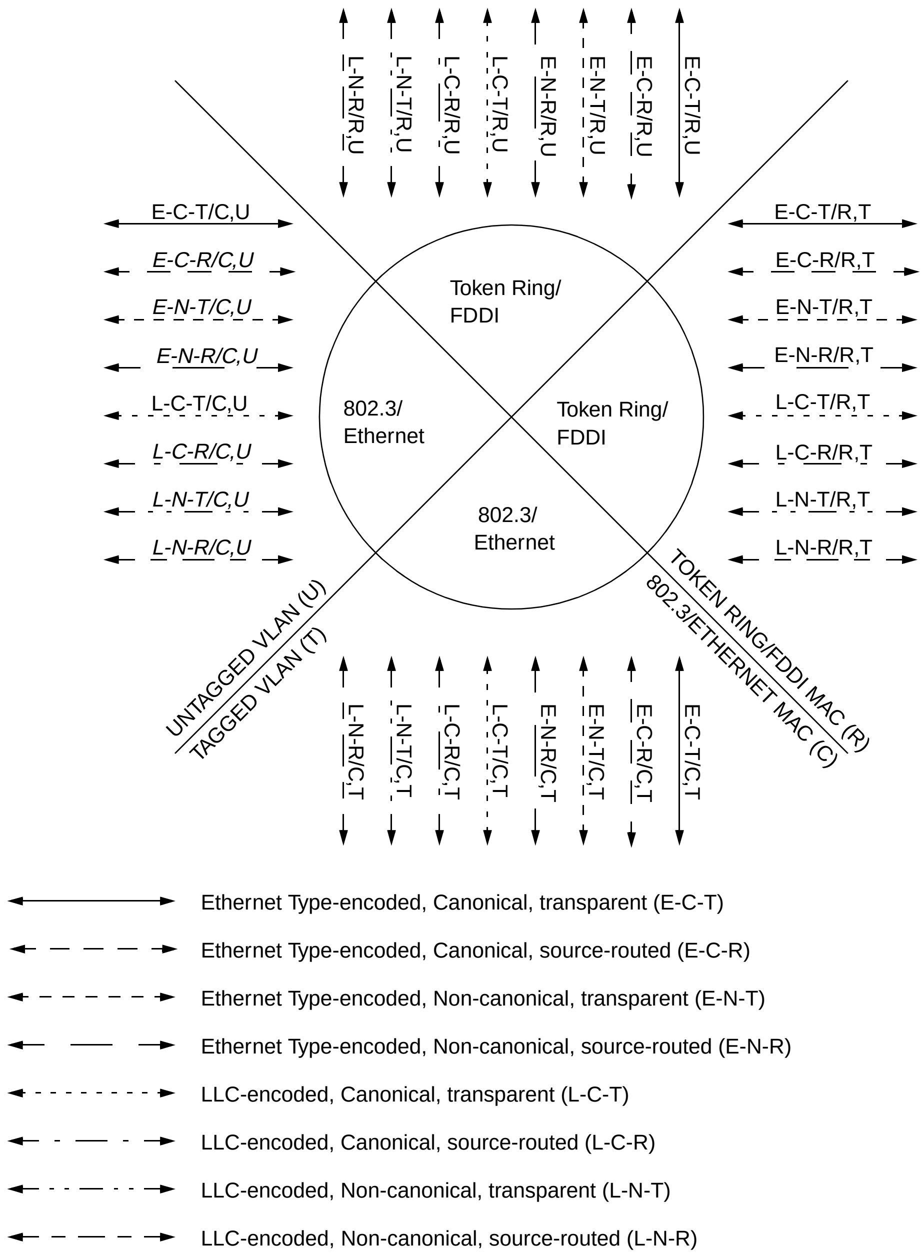


Figure C-1—Services and environments

Figure C-2 illustrates the heterogeneous Bridging functions involved. In this diagram, the bridging functions are labelled H, Q or Q+H depending upon whether the function involves ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 frame translation, 802.1Q VLAN-tagging/untagging/tag translation, or a combination of 802.1Q VLAN-tagging/untagging/tag translation and ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 frame translation.

NOTE 2—Figure C-2 is not intended to represent a real LAN, simply to illustrate the various Bridging functions.

In both diagrams, the frame formats involved are identified by three initial letters that identify the service provided, from the list of 8 services above. The fourth letter indicates the MAC method that carries the frame (C or R), and the fifth letter indicates the type of VLAN (U or T).

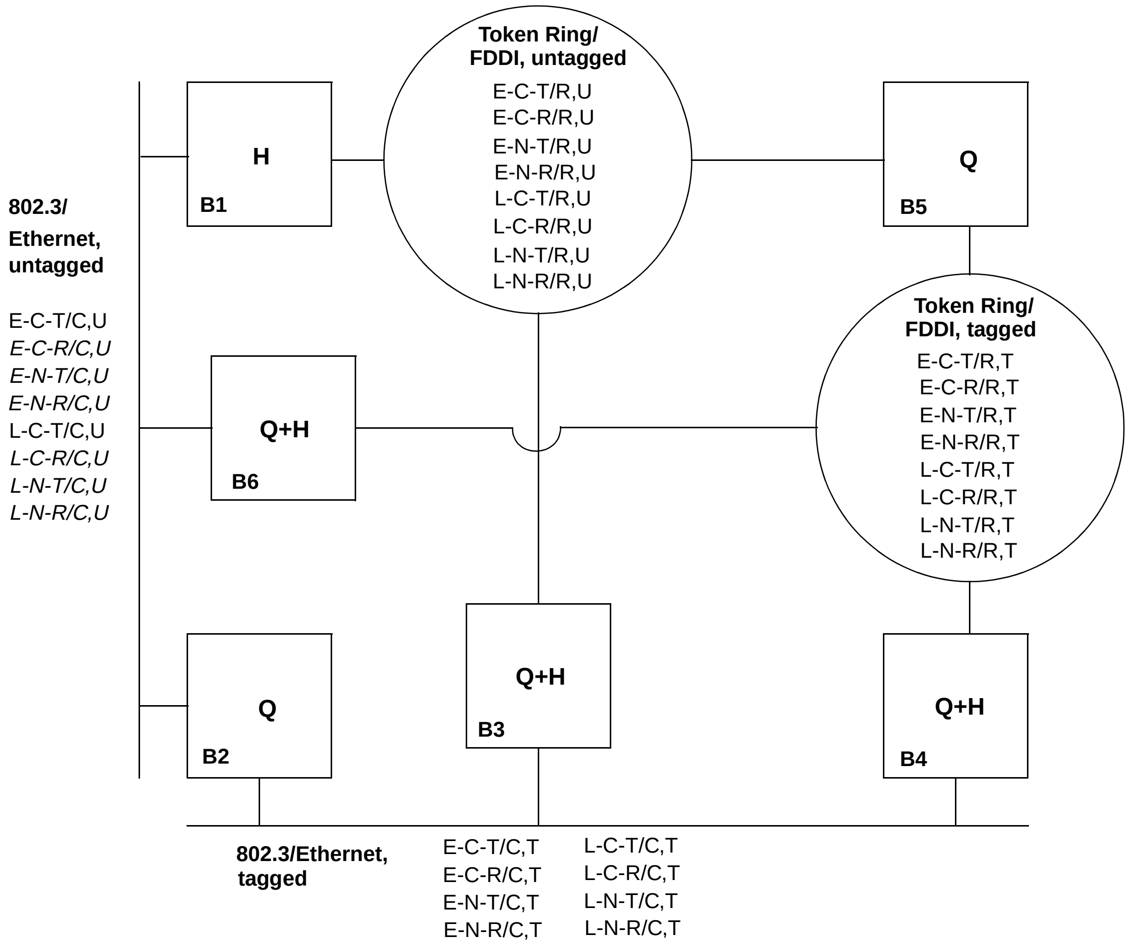


Figure C-2—Heterogeneous Bridging functions

C.2 Bridging functions

C.2.1 Bridging function B1

This function bridges between heterogeneous untagged VLAN environments. The following frame translations are involved:

a) ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation/decapsulation for frames carrying Ethernet Type-encoded information;
b) ISO/IEC 15802-3 conversion for frames carrying LLC-encoded information.
c) Any requirements for translation from Canonical to Non-canonical address format, or vice versa, must be met if communication is to be maintained between end stations separated by this Bridging function.
d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

C.2.2 Bridging function B2

This function involves VLAN entry (I to T) and exit (T to I); it bridges between untagged and tagged 802.3/Ethernet environments. The frame translations involved are as follows:

a) Insertion of Ethernet-encoded tag headers on VLAN entry;
b) Removal of Ethernet-encoded tag headers on VLAN exit;
c) Translation of Non-canonical MAC Addresses to Canonical on VLAN exit;
d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

NOTE—VLAN entry in Non-canonical format does not occur, as the native representation on 802.3/Ethernet is Canonical format. VLAN exit of Non-canonical information can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses to their Canonical format.

C.2.3 Bridging function B3

This function involves VLAN entry and exit; it bridges between tagged 802.3/Ethernet and untagged Ring environments. The following frame translations are involved:

a) For untagged Ethernet Type-encoded information on Token Ring/FDDI (VLAN entry): Removal of ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation, and insertion of Ethernet-encoded tag header;
b) For tagged Ethernet Type-encoded information on 802.3/Ethernet (VLAN exit): Removal of tag header and insertion of ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation;
c) For VLAN entry/exit with frames carrying LLC-encoded information: Insertion/removal of Ethernet-encoded tag header;
d) Translation of MAC Addresses to the format appropriate for the destination Ring on VLAN exit;
e) Any source-routing information present in the frame is preserved; the Token Ring/FDDI RIF is copied into the E-RIF in the Ethernet-encoded tag header on VLAN entry (with the CFI/NCFI flags set appropriately), and copied back on exit.

NOTE—VLAN entry (tagged 802.3/Ethernet from untagged Token Ring/FDDI) in Canonical format normally occurs only from ISO/IEC 9314-2, as the native representation on ISO/IEC 8802-5 is Non-canonical format, and for ISO/IEC 9314-2 is Canonical format. Hence, VLAN exit in Canonical format onto ISO/IEC 8802-5 can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Canonical on 802.3/Ethernet to Non-canonical on ISO/IEC 8802-5. Similarly, VLAN exit in Non-canonical format onto ISO/IEC 9314-2 can occur only if the Bridge is capable of converting the frame from Non-canonical on 802.3/Ethernet to Canonical on ISO/IEC 9314-2.

C.2.4 Bridging function B4

This function bridges between tagged 802.3/Ethernet and Ring environments. The following frame translations are involved:

a) For tagged Ethernet Type-encoded information on Token Ring/FDDI to 802.3/Ethernet: Removal of ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation, and conversion of the tag header to the Ethernet-encoded form;
b) For tagged Ethernet Type-encoded information on 802.3/Ethernet to Token Ring/FDDI: ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation, and conversion of the tag header to the SNAP-encoded form;
c) For tagged frames carrying LLC-encoded information: conversion of the tag header between the SNAP-encoded and Ethernet-encoded forms;

d) Any source-routing information is preserved between these environments by copying between the Token Ring/FDDI RIF and the E-RIF in the Ethernet-encoded tag header.

NOTE 1—This Bridging function is not required to modify the format of embedded MAC Addresses.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.2.5 Bridging function B5

This function involves VLAN entry and exit; it bridges between tagged and untagged Ring environments. The frame translations involved are as follows:

a) Insertion of SNAP-encoded tag header on VLAN entry;
b) Removal of SNAP-encoded tag header on VLAN exit;
c) Translation of MAC Addresses to the format appropriate for the destination Ring on VLAN exit;
d) Any source-routing information present in the frame is preserved.

NOTE 1—VLAN entry in Canonical format normally occurs only from ISO/IEC 9314-2, as the native representation on ISO/IEC 8802-5 is Non-canonical format, and for ISO/IEC 9314-2 is Canonical format. Hence, VLAN exit in Canonical format onto ISO/IEC 8802-5 can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Canonical on 802.3/Ethernet to Non-canonical on ISO/IEC 8802-5. Similarly, VLAN exit in Non-canonical format onto ISO/IEC 9314-2 can occur only if the Bridge is capable of converting the frame from Non-canonical on 802.3/Ethernet to Canonical on ISO/IEC 9314-2.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.2.6 Bridging function B6

This function involves VLAN entry and exit; it bridges between untagged 802.3/Ethernet and tagged Ring environments. The following frame translations are involved:

a) For untagged Ethernet Type-encoded information on 802.3/Ethernet (VLAN entry): ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation, and insertion of SNAP-encoded tag header;
b) For tagged Ethernet Type-encoded information on Token Ring/FDDI (VLAN exit): Removal of SNAP-encoded tag header and removal of ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 encapsulation;
c) For VLAN entry/exit with frames carrying LLC-encoded information: Insertion/removal of SNAP-encoded tag header;
d) Any source-routed traffic cannot be relayed between these environments, as there is no representation for source-routing information in untagged frames on 802.3/Ethernet LANs.

NOTE 1—VLAN entry in Non-canonical format does not occur, as the native representation on 802.3/Ethernet is Canonical format. VLAN exit of Non-canonical format can occur only if the Bridge is capable of translating the representation of embedded MAC Addresses; i.e., of converting the frame from Non-canonical format to Canonical format on 802.3/Ethernet.
NOTE 2—In FDDI LANs, source-routing information may be present either in an E-RIF within the tag header or in the normal position for a source-routed frame.

C.3 Frame formats

The following abbreviations are used in the descriptions of the frame formats in this annex, with the following meanings:

AC	Access Control field—in Token Ring frames only (see ISO/IEC 15802-5 and Note below)
RCI	Ring Control Information—AC (if present) plus FC fields
DA	Destination MAC Address
SA	Source MAC Address
PT	Ethernet Protocol Type
SPT	SNAP-encoded Ethernet Protocol Type (C.6.1)
TPID	Tag Protocol ID (9.3.1)
ETPID	Ethernet-encoded TPID (9.3.1.1)
STPID	SNAP-encoded TPID (9.3.1.2)
TCI	Tag Control Information (9.3.2)
CFI	Canonical Format Indicator (9.3.2.2)
NCFI	Non-canonical Format Indicator (9.3.3.5)
C	Canonical
N	Non-canonical
R	E-RIF present
VID	VLAN Identifier (9.3.2.3)
Len	IEEE Std 802.3-style Length/Type field (C.6.2)
LLC	LLC addressing and control information, as defined in ISO/IEC 15802-2
RIF	Source-Routing Information Field (C.6.4)
E-RIF	Embedded RIF (9.3.3, C.6.4)
C-Data	MAC user data in which any embedded MAC Addresses are in Canonical format (C.6.3)
N-Data	MAC user data in which any embedded MAC Addresses are in Non-canonical format
PAD	Padding (C.6.5)
FCS	Frame Check Sequence

In C.3.2, the possible frame formats are categorized by service type; in C.3.3 they are categorized by bearer MAC method and tagging method.

NOTE—The text in this annex makes the generalization of treating the FC fields in 8802-5 and FDDI as if they are the same, in order to simplify the descriptions as much as possible. In reality, there are detailed differences between FC fields in the two MAC methods. When translating between 8802-5 and FDDI, the most likely behavior is to propagate the “LLC frame” indication and the User Priority field between the FC octets on input and output.

C.3.1 Structure of the tagged frame

Figure C-3 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information using 8802-5 Token Ring MAC methods.

Figure C-4 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information using FDDI MAC methods. Two forms of tagged frame are shown:

a) The *source-routed form*, in which the frame carries a RIF in the normal position, following the source MAC Address. This form can only be used on FDDI LANs that support source routing; and
b) The *transparent form*, in which an E-RIF is present in the tag header if the frame carries Non-canonical or source-routed information.

Figure C-5 illustrates the frame formats used for carrying tagged Ethernet Type-encoded information and LLC-encoded information on 802.3/Ethernet MAC methods.

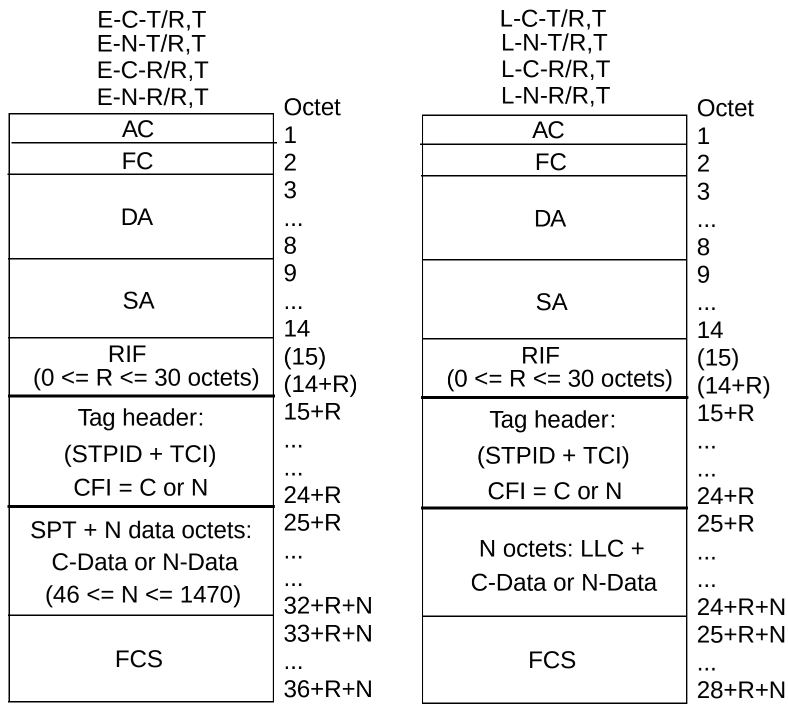


Figure C-3—Tagged frames on 8802-5 Token Ring LANs

As can be seen from these diagrams, the major differences between the tagged frame formats in 802.3/Ethernet and Token Ring/FDDI MAC methods are

c) The presence/absence of RCI (Ring Control Information);
d) The position of the RIF and E-RIF fields;
e) The encoding used to carry the Tag Protocol Identifier (2 octets for ETPID vs. 8 octets for STPID);
f) The encoding used to carry Ethernet Protocol Types (2 octets for PT vs. 8 octets for SPT);
g) The presence/absence of the Length/Type field;
h) The presence/absence of the PAD field.

The diagrams also illustrate the similarities between:

i) The format of tagged frames on 8802-5 and the source-routed form of tagged frames on FDDI;
j) The format of tagged frames on 802.3/Ethernet and the transparent form of tagged frames on FDDI.

C.3.2 Frame formats by service type

C.3.2.1 Frame formats for Ethernet Type-encoded service

C.3.2.1.1 Ethernet Type-encoded, Canonical, transparent

E-C-T/C,U:	DA, SA, PT, C-Data, FCS
E-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), PT, C-Data, FCS
E-C-T/R,U:	RCI, DA, SA (RII reset), SPT, C-Data, FCS
E-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), SPT, C-Data, FCS

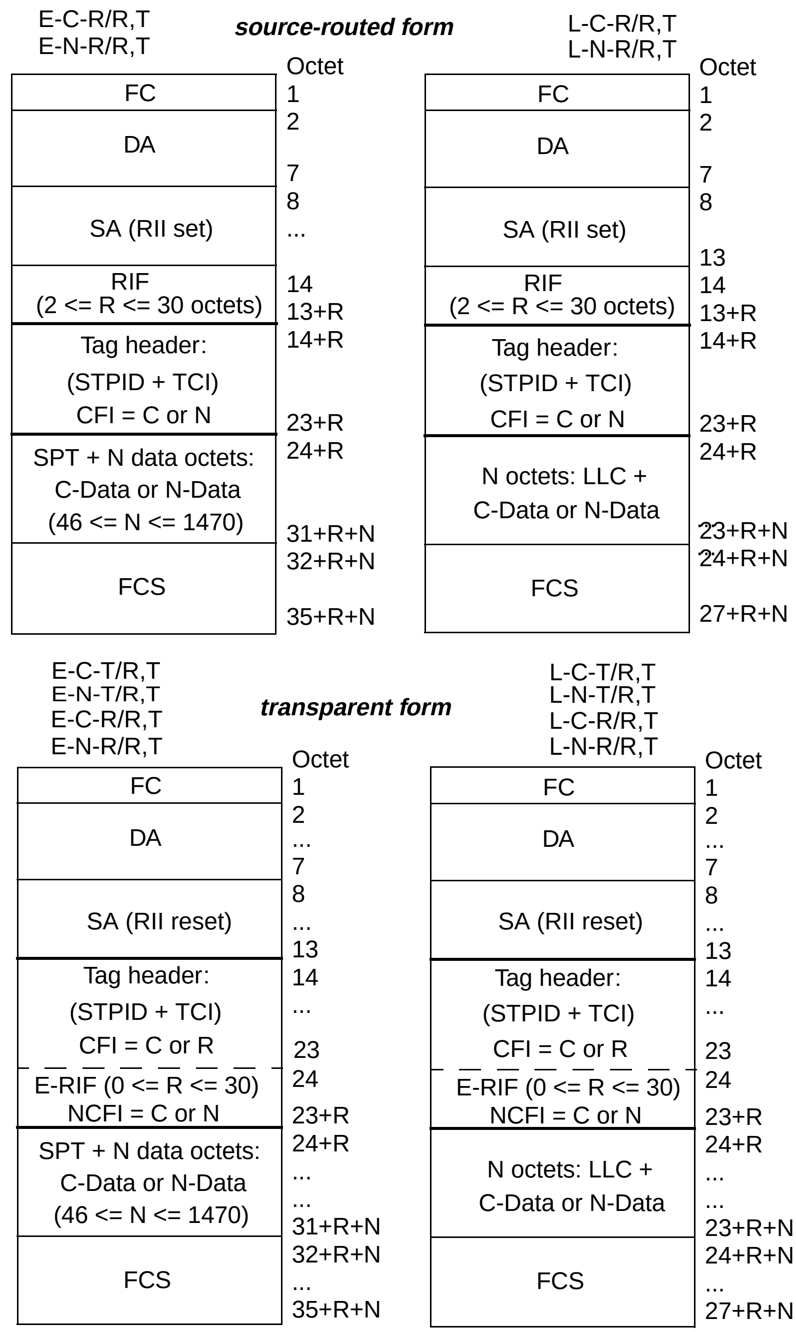


Figure C-4—Tagged frames on FDDI LANs

C.3.2.1.2 Ethernet Type-encoded, Canonical, source-routed

E-C-R/C,U:	No representation possible
E-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=C), C-Data, FCS
E-C-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, C-Data, FCS
E-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), SPT, C-Data, FCS *(source-routed form)*
E-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), SPT, C-Data, FCS *(transparent form)*

C.3.2.1.3 Ethernet Type-encoded, Non-canonical, transparent

E-N-T/C,U: No representation possible
E-N-T/C,T: DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-T/R,U: RCI, DA, SA (RII reset), SPT, N-Data, FCS
E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI =N), SPT, N-Data, FCS
(8802-5 Token Ring form)
E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), SPT, N-Data, FCS
(FDDI form)

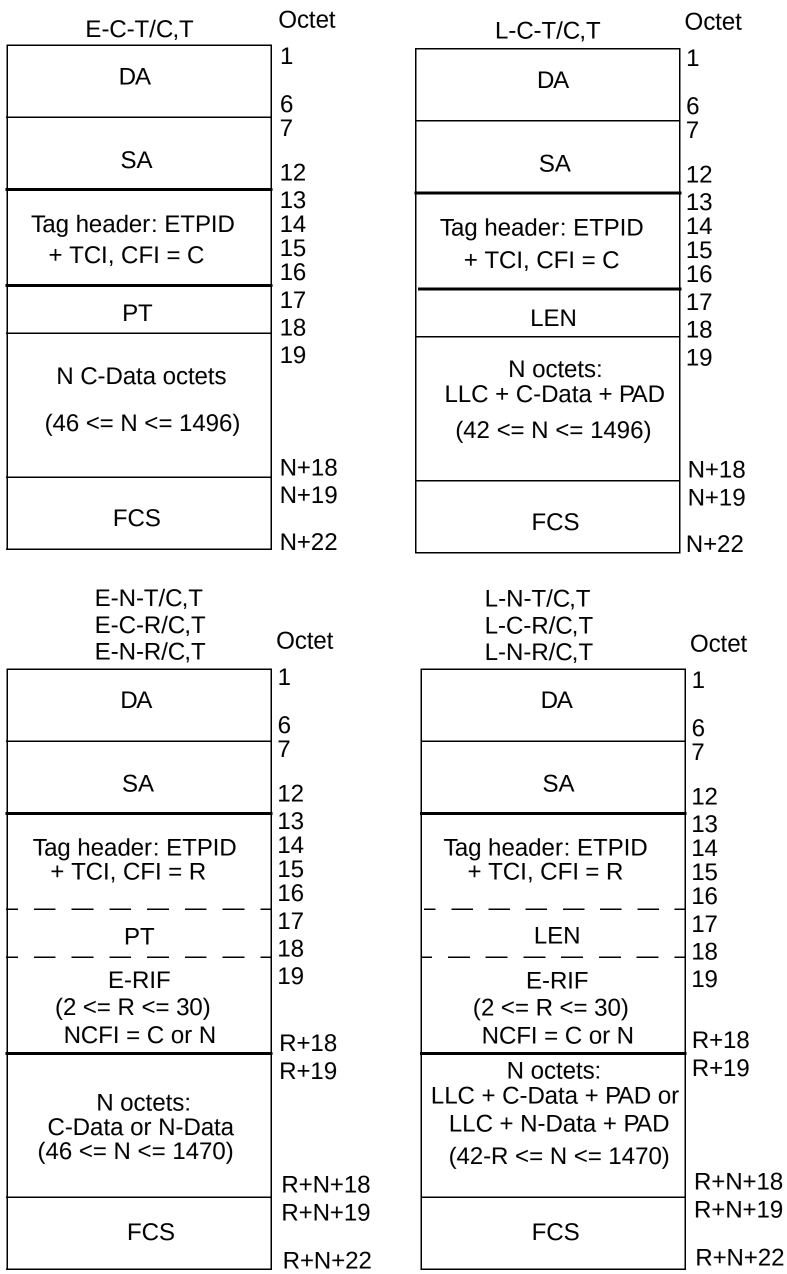


Figure C-5—Tagged frames on 802.3/Ethernet LANs

C.3.2.1.4 Ethernet Type-encoded, Non-canonical, source-routed

E-N-R/C,U: No representation possible
E-N-R/C,T: DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-R/R,U: RCI, DA, SA (RII set), RIF, SPT, N-Data, FCS
E-N-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), SPT, N-Data, FCS *(source-routed form)*
E-N-R/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), N-Data, FCS *(transparent form)*

C.3.2.2 Frame formats for LLC-encoded service

C.3.2.2.1 LLC-encoded, Canonical, transparent

L-C-T/C,U: DA, SA, LEN, LLC, C-Data, PAD, FCS
L-C-T/C,T: DA, SA, ETPID, TCI (CFI=C), LEN, LLC, C-Data, PAD, FCS
L-C-T/R,U: RCI, DA, SA (RII reset), LLC, C-Data, FCS
L-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=C), LLC, C-Data, FCS

C.3.2.2.2 LLC-encoded, Canonical, source-routed

L-C-R/C,U: No representation possible
L-C-R/C,T: DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=C), LLC, C-Data, PAD, FCS
L-C-R/R,U: RCI, DA, SA (RII set), RIF, LLC, C-Data, FCS
L-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), LLC, C-Data, FCS *(source-routed form)*
L-C-R/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), LLC, C-Data, FCS *(transparent form)*

C.3.2.2.3 LLC-encoded, Non-canonical, transparent

L-N-T/C,U: No representation possible
L-N-T/C,T: DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-T/R,U: RCI, DA, SA (RII reset), LLC, N-Data, FCS
L-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=N), LLC, N-Data, FCS *(8802-5 Token Ring form)*
L-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(FDDI form)*

C.3.2.2.4 LLC-encoded, Non-canonical, source-routed

L-N-R/C,U: No representation possible
L-N-R/C,T: DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-R/R,U: RCI, DA, SA (RII set), RIF, LLC, N-Data, FCS
L-N-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), LLC, N-Data, FCS *(source-routed form)*
L-N-R/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(transparent form)*

C.3.3 Frame formats by MAC method type and tagging method

C.3.3.1 Frame formats for 802.3/Ethernet MAC methods

C.3.3.1.1 802.3/Ethernet, untagged

E-C-T/C,U:	DA, SA, PT, C-Data, FCS
E-C-R/C,U:	No representation possible
E-N-T/C,U:	No representation possible
E-N-R/C,U:	No representation possible
L-C-T/C,U:	DA, SA, LEN, LLC, C-Data, PAD, FCS
L-C-R/C,U:	No representation possible
L-N-T/C,U:	No representation possible
L-N-R/C,U:	No representation possible

C.3.3.1.2 802.3/Ethernet, tagged

E-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), PT, C-Data, FCS
E-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=C), C-Data, FCS
E-N-T/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
E-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), PT, E-RIF (NCFI=N), N-Data, FCS
L-C-T/C,T:	DA, SA, ETPID, TCI (CFI=C), LEN, LLC, C-Data, PAD, FCS
L-C-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=C), LLC, C-Data, PAD, FCS
L-N-T/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS
L-N-R/C,T:	DA, SA, ETPID, TCI (CFI=R), LEN, E-RIF (NCFI=N), LLC, N-Data, PAD, FCS

C.3.3.2 Frame formats for Token Ring/FDDI MAC methods

C.3.3.2.1 Token Ring/FDDI, untagged

E-C-T/R,U:	RCI, DA, SA (RII reset), SPT, C-Data, FCS
E-C-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, C-Data, FCS
E-N-T/R,U:	RCI, DA, SA (RII reset), SPT, N-Data, FCS
E-N-R/R,U:	RCI, DA, SA (RII set), RIF, SPT, N-Data, FCS
L-C-T/R,U:	RCI, DA, SA (RII reset), LLC, C-Data, FCS
L-C-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, C-Data, FCS
L-N-T/R,U:	RCI, DA, SA (RII reset), LLC, N-Data, FCS
L-N-R/R,U:	RCI, DA, SA (RII set), RIF, LLC, N-Data, FCS

C.3.3.2.2 Token Ring/FDDI, tagged

E-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), SPT, C-Data, FCS
E-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), SPT, C-Data, FCS *(source-routed form)*
E-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), SPT, C-Data, FCS *(transparent form)*
E-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=N), SPT, N-Data, FCS *(8802-5 Token Ring form)*
E-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), SPT, N-Data, FCS *(FDDI form)*
E-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), SPT, N-Data, FCS *(source-routed form)*
E-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), N-Data, FCS *(transparent form)*

L-C-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=C), LLC, C-Data, FCS
L-C-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=C), LLC, C-Data, FCS *(source-routed form)*
L-C-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=C), LLC, C-Data, FCS *(transparent form)*
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=N), LLC, N-Data, FCS *(8802-5 Token Ring form)*
L-N-T/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(FDDI form)*
L-N-R/R,T:	RCI, DA, SA (RII set), RIF, STPID, TCI (CFI=N), LLC, N-Data, FCS *(source-routed form)*
L-N-R/R,T:	RCI, DA, SA (RII reset), STPID, TCI (CFI=R), E-RIF (NCFI=N), LLC, N-Data, FCS *(transparent form)*

C.4 Procedures for tagging, untagging, and relaying tagged frames

The formal definition of the procedures whereby tag headers are added and removed, and tagged frames are relayed are embodied in Clauses 7 and 8. This informal description is included in order to add clarity to the formal definition of the process.

C.4.1 Tagging

The following subclauses describe the translations that are performed when an untagged frame is relayed in tagged form.

C.4.1.1 MAC header information

The RCI, DA, SA and RIF fields (if supported in the source frame and/or destination MAC methods) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in ISO/IEC 15802-3. This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC methods;
c) Preservation of the RIF field, if present; either in its conventional position (Token Ring/FDDI to Token Ring/FDDI, source-routed form) or within the tag header (Token Ring/FDDI to 802.3/Ethernet or FDDI, transparent form).

NOTE—The ability of the tag header to carry source-routing information across 802.3/Ethernet LANs does not imply a requirement on the part of a pure 802.3/Ethernet Bridge to support source routing. This capability is provided simply to allow traffic that originates in, and is destined for, a source-routed environment to transit as tagged traffic across a non-source-routed environment. Similarly, this capability allows source-routed traffic to transit an FDDI network that is otherwise unable to support source routing.

C.4.1.2 Tag header insertion

The tag header is inserted immediately following the SA field (if no RIF is present in the destination frame) or immediately following the RIF field (if RIF is present in the destination frame). The header contains

a) An Ethernet-encoded TPID (destination MAC method is 802.3/Ethernet) or a Snap-encoded TPID (destination MAC method is Token Ring/FDDI);
b) A TCI field, as follows:
    1) The user_priority field is set in accordance with the procedure described in ISO/IEC 15802-3;

2) The CFI flag, indicating C/N (8802-5 Token Ring, and source-routed FDDI MAC methods), or C/[RIF present] (802.3/Ethernet and transparent FDDI MAC methods), in accordance with the format of the MAC user data;
3) The VID field is set to the VID of the VLAN to which the source frame belongs.

c) An E-RIF field, immediately following the Length/Type field (802.3/Ethernet and transparent FDDI MAC methods), if the frame is carrying Non-canonical data and/or source-routing information. The NCFI in the RIF indicates C or N, in accordance with the format of the MAC user data.

C.4.1.3 Ethernet Type-encoded data

If the MAC user data carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in ISO/IEC 15802-3 and ISO/IEC 11802-5, and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the procedures described in ISO/IEC 15802-3 and ISO/IEC 11802-5.

C.4.1.4 FCS

When tagging a frame and performing the attendant field translations, it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in ISO/IEC 15802-3, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique, and other techniques for preserving FCS integrity, are discussed in ISO/IEC 15802-3, Annex G.

C.4.2 Untagging

The following subclauses describe the frame translations that are performed when a received tagged frame is relayed in untagged format.

C.4.2.1 MAC header information

The RCI, DA, SA and RIF or E-RIF fields (if supported in the source frame and/or destination MAC methods) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in ISO/IEC 15802-3. This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC method;
c) Preservation of any source-routing information carried in the source frame, if present, and if the destination MAC method is a Token Ring/FDDI LAN that supports source routing. (If the source MAC method is 802.3/Ethernet or transparent FDDI and the tag header carries an E-RIF in which the RT field indicates a transparent frame, then the E-RIF is not considered to be carrying any source-routing information.)

C.4.2.2 Tag header

The tag header is removed.

C.4.2.3 Ethernet Type-encoded data

If the frame carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in ISO/IEC 15802-3 and ISO/IEC 11802-5,

and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the with the procedures described in ISO/IEC 15802-3 and ISO/IEC 11802-5.

C.4.2.4 Address translation

If the CFI/NCFI information in the tagged frame indicates that embedded addresses are being carried in a format inappropriate to the destination MAC method, then it is necessary either to translate the addresses from C to N or vice versa, or to discard the frame if such translation is not supported by the Bridge.

C.4.2.5 FCS

When removing a frame's VLAN tag and performing the attendant field translations, it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in ISO/IEC 15802-3, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique, and other techniques for preserving FCS integrity, are discussed in ISO/IEC 15802-3, Annex G.

C.4.3 Relaying tagged frames

The following subclauses describes the frame translations that are performed when a received tagged frame is relayed in tagged format.

C.4.3.1 MAC header information

The RCI, DA, and SA (if supported in the source frame and/or destination frame formats) are translated from their representation in the source frame into the equivalent representation in the destination frame in accordance with the procedures described in ISO/IEC 15802-3.

For source-routed Token Ring/FDDI to 802.3/Ethernet or transparent FDDI, the RIF field (if present) is translated into the E-RIF field of the destination frame.

For relay between source-routed Token Ring/FDDI environments, the RIF (if present) is copied into the RIF field of the destination frame.

For 802.3/Ethernet or transparent FDDI to source-routed Token Ring/FDDI, the E-RIF field, if present, is translated into the RIF of the destination frame, with the NCFI bit reset, unless the E-RIF indicates that the frame is a transparent frame, in which case, the E-RIF is discarded.

This will result in

a) Preservation of the AC (Token Ring to Token Ring only) and FC fields (Token Ring/FDDI to Token Ring/FDDI only);
b) Translation of the DA and SA into their equivalent representation in the destination MAC method;
c) Preservation of any information carried in the E-RIF or RIF field, if present and if it carries source-routing information.

C.4.3.2 Tag header

If the source and destination MAC methods differ, the tag header is modified as follows:

a) The TPID field is set in accordance with the destination MAC method. An Ethernet-encoded TPID is used where the destination MAC method is 802.3/Ethernet; a Snap-encoded TPID is used where the destination MAC method is Token Ring/FDDI;
b) The information carried in the User Priority and VID fields in the TCI are copied unchanged into the destination frame's TCI.
c) If the source and destination MAC methods are of the same type, then the CFI (and RIF, if present in 802.3/Ethernet) are copied unchanged into the destination tag header.
d) If the source and destination MAC methods differ, then the CFI information in the source tag header is translated into the format appropriate for the destination tag header.

C.4.3.3 Ethernet Type-encoded data

If the frame carries Ethernet Type-encoded data, i.e., the protocol identifier is an Ethernet Type value or an Ethernet Type value that has been SNAP encoded as described in ISO/IEC 15802-3 and ISO/IEC 11802-5, and if the frame is being relayed between differing MAC methods (802.3/Ethernet to or from Token Ring/FDDI), then the data is translated from its source format to the format appropriate to the destination MAC method in accordance with the with the procedures described in ISO/IEC 15802-3 and ISO/IEC 11802-5.

C.4.3.4 FCS

When relaying tagged frames, if it is necessary to perform any attendant field translations, then it is necessary to recompute the Frame Check Sequence (FCS) field of the tagged frame. As stated in ISO/IEC 15802-3, 6.3.7, the Bridge shall not introduce additional undetected frame errors as a result of such FCS recomputation.

NOTE—Where necessary in order to preserve the protection afforded by the original FCS, it is possible to incrementally compute the new FCS value, based on the original FCS, adjusted for the new fields added and the frame length. This technique and other techniques for preserving FCS integrity are discussed in ISO/IEC 15802-3, Annex G.

C.4.4 Padding and frame size considerations

C.4.4.1 Treatment of PAD fields in IEEE Std 802.3 frames

The minimum frame size constraint placed on 802.3/Ethernet frames requires frames to carry zero or more pad octets following the MAC client data, in order to ensure that no frame of total length less than 64 octets is transmitted on the medium. This means that frames whose overall length would otherwise be less than 64 octets in length have (64-len) octets of padding added after the MAC client data, where len is the size of the frame before padding.

When tagged frames are transmitted by a Bridge on an IEEE Std 802.3 MAC, there are two permissible approaches (7.2):

a) Keep the minimum frame size generated by the Bridge equal to 64 octets. This implies that the number of pad octets in a received untagged IEEE Std 802.3 frame would be reduced by up to 4 octets when that frame was tagged;
b) Adopt a minimum tagged frame length of 68 octets. This implies that the number of pad octets in a received untagged IEEE Std 802.3 frame would not be adjusted when tagging such frames; equally, if subsequently untagged, no pad adjustment would be necessary before transmission on 802.3/Ethernet.

There is a similar choice to be made in end stations that generate tagged frames:

c) In some existing implementations, the decision as to whether pad octets are needed will be made at a point where it is impractical to distinguish between tagged and untagged frames. In these cases, the end station will use a minimum frame size of 64 octets for all frames;
d) In other cases, the padding decision will be taken at a point before it is known whether the frame will be transmitted tagged or untagged. In these cases, the end station will use a minimum tagged frame size of 68 octets, and a minimum of 64 octets for untagged frames.

The above approaches are all consistent with the IEEE Std 802.3 frame specification, as amended by IEEE Std 802.3ac-1998.

The implication of this is that, for correct operation on 802.3/Ethernet, all devices have to be capable of correctly handling tagged frames of less than 68 octets in length (C.4.4.3).

C.4.4.2 Maximum PDU size

VLAN tagging of an untagged frame, or relaying frames in tagged frame format, can result in an increase in the length of the original frame. If transmission of a given frame in tagged frame format through a given destination Port would result in violation of the maximum PDU size for the destination MAC method, the Bridge discards the frame for that destination Port.

NOTE—Violation of the maximum PDU sizes for destination MAC methods can produce undefined results in Bridged LANs that contain devices that adhere strictly to these maxima, or in MAC methods where these maxima are inherently constrained by the operation of the MAC method itself (e.g., constrained by timing considerations in the MAC state machines).

IEEE Std 802.3ac-1998 defines an extension to the normal 802.3 maximum frame size for the specific purpose of accommodating the additional octets of the VLAN tag header. The example frame translations in this annex make use of this extension to the 802.3 frame size.

C.4.4.3 Minimum PDU size

VLAN untagging of a tagged frame results in the original frame decreasing in length.

Where the destination MAC is CSMA/CD:

a) If untagging a given frame would result in violation of the minimum frame length requirements of CSMA/CD, the Bridge is required to adjust the PAD field to ensure that the frame length equals the minimum length of 64 octets (7.2 and C.4.4.1);
b) If a frame is transmitted in tagged frame format, the Bridge may adopt a minimum tagged frame length of either 64 or 68 octets, as an implementation option. If the latter is chosen, the Bridge adjusts the size of the PAD field on transmission for any tagged frame that is less than 68 octets in length (7.2, C.4.4.1).

C.5 Frame translations for different MAC methods

Examples of the frame translations that can occur when an untagged frame is translated into a tagged frame, and when tagged frames are relayed, are illustrated in the following clauses.

Subclauses C.5.1 and C.5.2 describe the translations that can occur when untagged frames on 802.3/Ethernet, and Token Ring/FDDI are translated into the tagged frame format. C.5.3 describes the translations that

can occur when a tagged frame is relayed between differing MAC methods in tagged format. In each subclause, the following cases are shown:

a) The untagged frame carried Ethernet Type-encoded information;
b) The untagged frame carried LLC-encoded information.

NOTE—In developing the example translations, the field sizes on 802.3/Ethernet have been calculated using the IEEE Std 802.3ac-1998 extension to the standard maximum frame size (normally 1518 octets). IEEE Std 802.3ac-1998 allows the maximum frame size to be extended by 4 octets for the specific purpose of accommodating the tag header.

C.5.1 Tagging of untagged 802.3/Ethernet frames

C.5.1.1 Ethernet Type-encoded information on 802.3/Ethernet LAN to tagged frame format

Figure C-6 illustrates the translation between an untagged Ethernet Type-encoded frame on 802.3/Ethernet (E-C-T/C,U) and a tagged frame on 802.3/Ethernet (E-C-T/C,T).

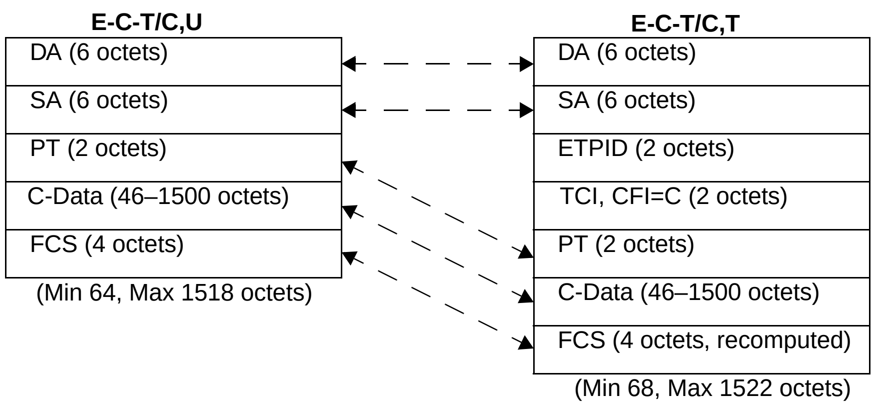


Figure C-6—Translation between E-C-T/C,U and E-C-T/C,T

The following translations are required in order to tag an E-C-T/C,U frame on 802.3/Ethernet:

a) The SA and DA fields are copied unchanged;
b) The ETPID and TCI are inserted, with CFI=C;
c) The PT and C-Data fields are copied unchanged;
d) The FCS is recomputed.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 4 octets.

Figure C-7 illustrates the translation between an untagged Ethernet Type-encoded frame on 802.3/Ethernet (E-C-T/C,U) and a tagged frame on Token Ring/FDDI (E-C-T/R,T).

The following translations are required in order to tag an E-C-T/C,U frame on a Token Ring/FDDI LAN:

e) The appropriate variant of the RCI field is added;
f) The DA and SA fields carry the same MAC Addresses as in the original frame;

NOTE—The meaning of the wording used in f) (and in other instances in this annex where this form of words is used) is that the MAC Addresses in the original and translated frames, when represented using the hexadecimal notation defined in Clause 5 of IEEE Standard 802, are the same.

g) The STPID and TCI are inserted, with CFI=C;
h) The PT is translated into the ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390-encoded form (SPT);
i) The C-Data field is copied unchanged;
j) The FCS is recomputed.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 17 octets for FDDI or 18 octets for Token Ring.

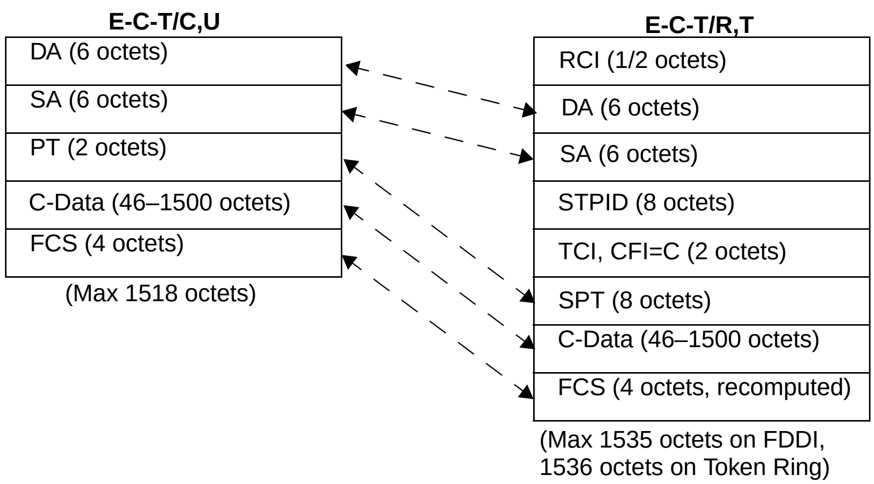


Figure C-7—Translation between E-C-T/C,U and E-C-T/R,T

NOTE—In translational (VLAN-unaware) bridging between 802.3/Ethernet and Ring LANs, an Ethernet Type-encoded frame increases in size by 7 octets on FDDI, and 8 octets on Token Ring.

Translations for E-C-R/C,U, E-N-T/C,U, and E-N-R/C,U to their equivalent tagged frame formats (E-C-R/C,T, E-N-T/C,T, and E-N-R/C,T on 802.3/Ethernet, and E-C-R/R,T, E-N-T/R,T and E-N-R/R,T on Token Ring/FDDI) cannot be shown, as there is no representation for such untagged frames on 802.3/Ethernet LANs. Similarly, translation of the tagged frames E-C-R/C,T, E-N-R/C,T, E-N-R/R,T, and E-C-R/R,T to untagged frames on 802.3/Ethernet is not possible, as it involves loss of the source-routing information. Translation of the remaining Non-canonical, transparent tagged frame formats into E-C-T/C,U is possible, but only if the Bridge is capable of translating Non-canonical data to its Canonical form.

C.5.1.2 LLC-encoded information on 802.3/Ethernet to tagged frame format

Figure C-8 illustrates the translation between an untagged frame on 802.3/Ethernet carrying LLC-encoded information (L-C-T/C,U) and a tagged frame on 802.3/Ethernet (L-C-T/C,T).

Tagging an L-C-T/C,U frame on 802.3/Ethernet LANs requires the following frame translations:

a) The DA and SA fields are copied unchanged;
b) Insert ETPID and TCI fields, with CFI=C;

c) Len, LLC and C-Data fields are copied unchanged;
d) The PAD may either be copied unchanged (giving a minimum tagged frame size of 68 octets), or reduced by up to 4 octets (giving a minimum tagged frame size of 64 octets), as an implementation option;

NOTE—If the actual length of the data portion of the frame is inconsistent with the value held in LEN, then this inconsistency is not corrected by the tagging process. This is done in order that protocols which generate such inconsistency, and which require that inconsistency to be maintained for their correct operation, are not broken by this aspect of tagging.

e) Recompute the FCS.

Removal of the tag involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 4 octets.

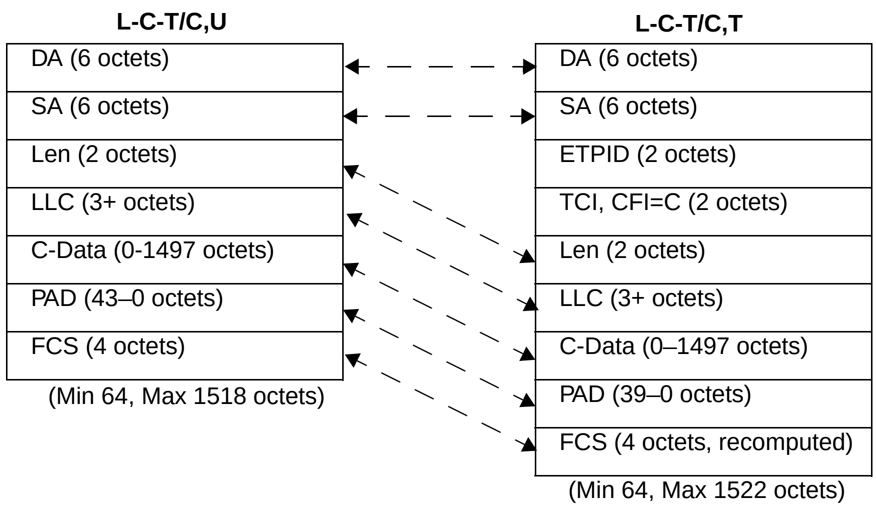


Figure C-8—Translation between L-C-T/C,U and L-C-T/C,T

Figure C-9 illustrates the translation between an untagged LLC-encoded frame on 802.3/Ethernet (L-C-T/C,U) and a tagged frame on Token Ring/FDDI (L-C-T/R,T).

Tagging in LLC-encoded format consists of the following frame translations:

f) The appropriate RCI field for the Ring MAC method concerned is added;
g) The DA and SA fields carry the same MAC Addresses as in the original frame;
h) Insert STPID and TCI fields, with CFI=C;
i) The Len field is removed;
j) Copy the LLC field unchanged;
k) The C-Data field is copied unchanged;
l) The PAD field is removed;
m) Recompute the FCS.

Removal of the tag (tagged frame to native 802.3/Ethernet frame) involves the reverse of this process.

This form of tagging causes the original frame size to be increased by 9 octets for FDDI or 10 octets for Token Ring.

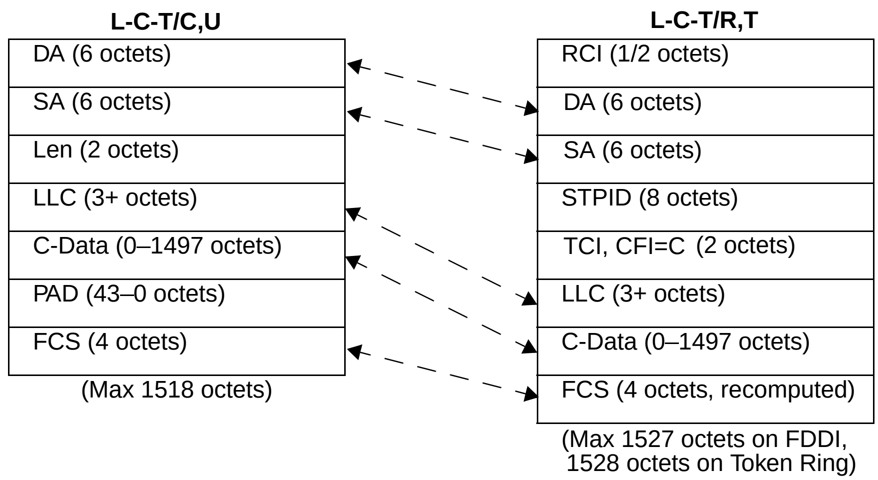


Figure C-9—Translation between L-C-T/C,U and L-C-T/R,T

NOTE—In translational (VLAN-unaware) bridging between 802.3/Ethernet and Ring LANs, an LLC-encoded frame reduces in size by 1 octet on FDDI, and does not change in length on Token Ring.

Translations for L-C-R/C,U, L-N-T/C,U, and L-N-R/C,U to their equivalent tagged frame formats (L-C-R/C,T, L-N-T/C,T, and L-N-R/C,T on 802.3/Ethernet, and L-C-R/R,T, L-N-T/R,T, and L-N-R/R,T on Token Ring/FDDI) cannot be shown, as there is no representation for such untagged frames on 802.3/Ethernet LANs. Similarly, translation of L-C-R/C,T, L-N-R/C,T, L-N-R/R,T, and L-C-R/R,T to untagged frames on 802.3/Ethernet is not possible, as it involves loss of the source-routing information. Translation of the remaining Non-canonical, transparent tagged frame formats into L-C-T/C,U is possible, but only if the Bridge is capable of translating Non-canonical data to its Canonical form.

C.5.2 Translation of untagged Token Ring/FDDI frames

C.5.2.1 Ethernet Type-encoded information on Token Ring/FDDI to tagged frame format

Figure C-10 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/FDDI (E-C-T/R,U, E-N-T/R,U, E-C-R/R,U or E-N-R/R,U) and a tagged frame on 802.3/Ethernet (E-C-T/C,T, E-N-T/C,T, E-C-R/C,T, or E-N-R/C,T).

Tagging requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Insert ETPID and TCI, with CFI=C (E-C-T/R,U) or R (all other frame types);
d) If the RII bit was set in the original frame, translate the RIF into the tag header E-RIF. For E-N-T/R,U, create a RIF with frame type = transparent. Set the E-RIF NCFI to C or N appropriately;
e) Translate the SPT into its corresponding PT;
f) Copy the Data field;
g) Recompute the FCS.

Removal of the tag involves the reverse of this process.

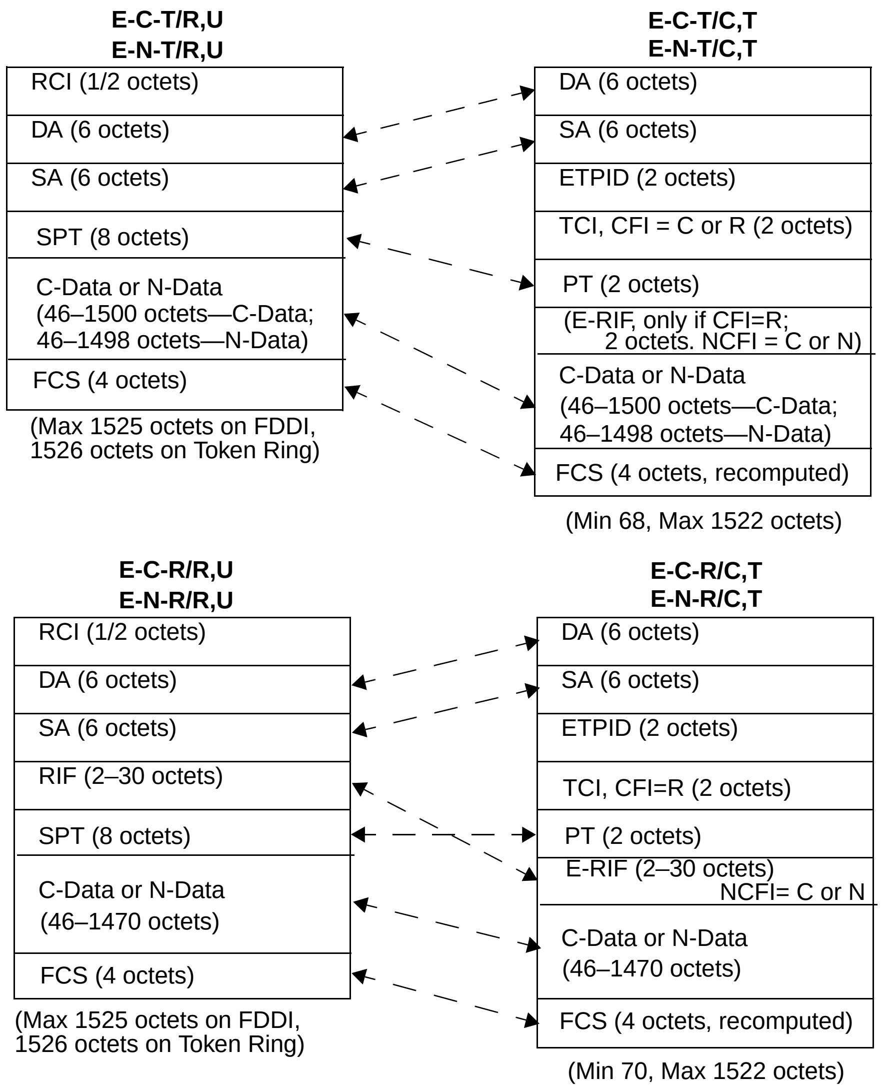


Figure C-10—Translation between E-X-X/R,U and E-X-X/C,T

NOTE 1—When removing the tag, if the CFI/NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be reduced by 3 or 4 octets.

Figure C-11 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/FDDI (E-C-T/R,U, E-N-T/R,U, E-C-R/R,U, or E-N-R/R,U) and a tagged frame on 8802-5 Token Ring (E-C-T/R,T, E-N-T/R,T, E-C-R/R,T, or E-N-R/R,T). Figure C-11 also illustrates the translation of E-C-T/R,U, E-C-R/R,U, and E-N-R/R,U to tagged frames on FDDI media, the latter two translations illustrating the source-routed form of the tagged frame on FDDI.

Tagging requires the following frame translations:

h) Copy the RCI field;
i) The DA and SA fields carry the same MAC Addresses as in the original frame, with RII in the same state as in the original frame;

j) Copy the RIF field if present (RII set);
k) Insert STPID and TCI, setting the CFI to N or C appropriately;
l) Copy the SPT field;
m) Copy the Data field;
n) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 2—When removing the tag, if the CFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 octets.

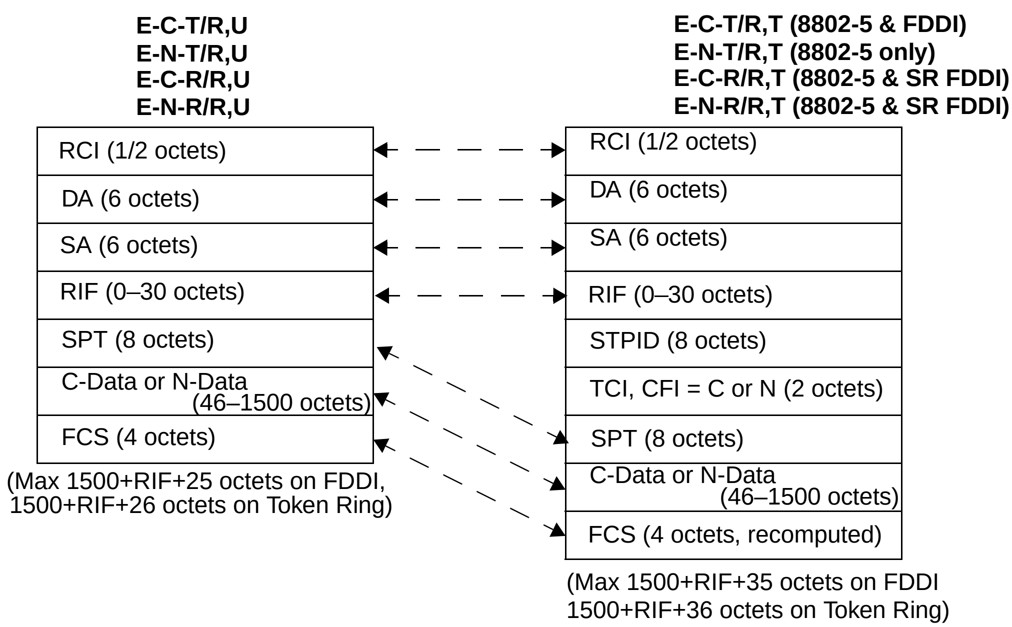


Figure C-11—Translation between E-X-X/R,U and E-X-X/R,T (8802-5 & SR FDDI)

Figure C-12 illustrates the translation between an untagged Ethernet Type-encoded frame on Token Ring/ FDDI (E-N-T/R,U, E-C-R/R,U or E-N-R/R,U) and a tagged frame on FDDI (E-N-T/R,T, E-C-R/R,T or E-N-R/R,T). Note that the translation of E-C-T/R,U to E-C-T/R,T was dealt with in Figure C-11.

Tagging requires the following frame translations:

o) Copy the RCI field;
p) The DA and SA fields carry the same MAC Addresses as in the original frame, but with RII reset regardless of its state in the original frame;
q) Insert STPID and TCI, setting the CFI to R;
r) Translate the RIF field if present (RII set in source frame) to the E-RIF, otherwise create an E-RIF with RT indicating a transparent frame. Set the NCFI to C or N appropriately;
s) Copy the SPT field;
t) Copy the Data field;
u) Recompute the FCS.

Removal of the tag (FDDI tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

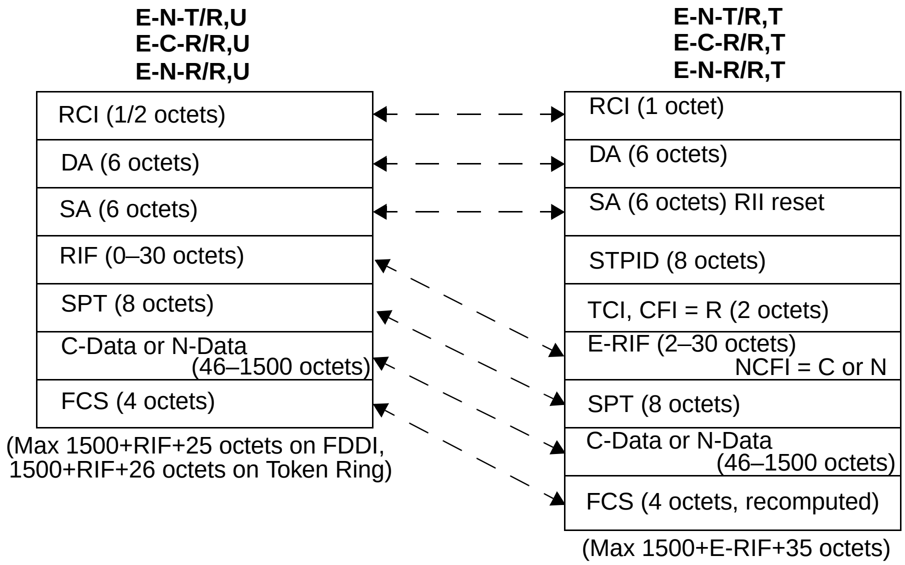


Figure C-12—Translation between E-X-X/R,U and E-X-X/R,T (transparent FDDI)

NOTE 3—When removing the tag, if the NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 or 12 octets.

C.5.2.2 LLC-encoded information on Token Ring/FDDI to tagged frame format

Figure C-13 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-C-T/R,U, L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 802.3/Ethernet (L-C-T/C,T, L-N-T/C,T, L-C-R/C,T, or L-N-R/C,T).

Tagging requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Insert ETPID and TCI, with CFI=C (L-C-T/R,U) or R (all other frame types);
d) If the RII bit was set in the original frame, translate the RIF into the tag header E-RIF. For L-N-T/R,U, create an E-RIF with frame type = transparent. Set the E-RIF NCFI to C or N appropriately;
e) Insert the Len field, with value equal to the number of LLC+Data octets;
f) Copy the LLC field;
g) Copy the Data field;
h) PAD field is inserted if Len is less than 46 (if a minimum tagged frame size of 68 is implemented) or if less than 42 (if a minimum tagged frame size of 64 is implemented);
i) Recompute the FCS.

Removal of the tag involves the reverse of this process.

NOTE 1—When removing the tag, if the CFI/NCFI information indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

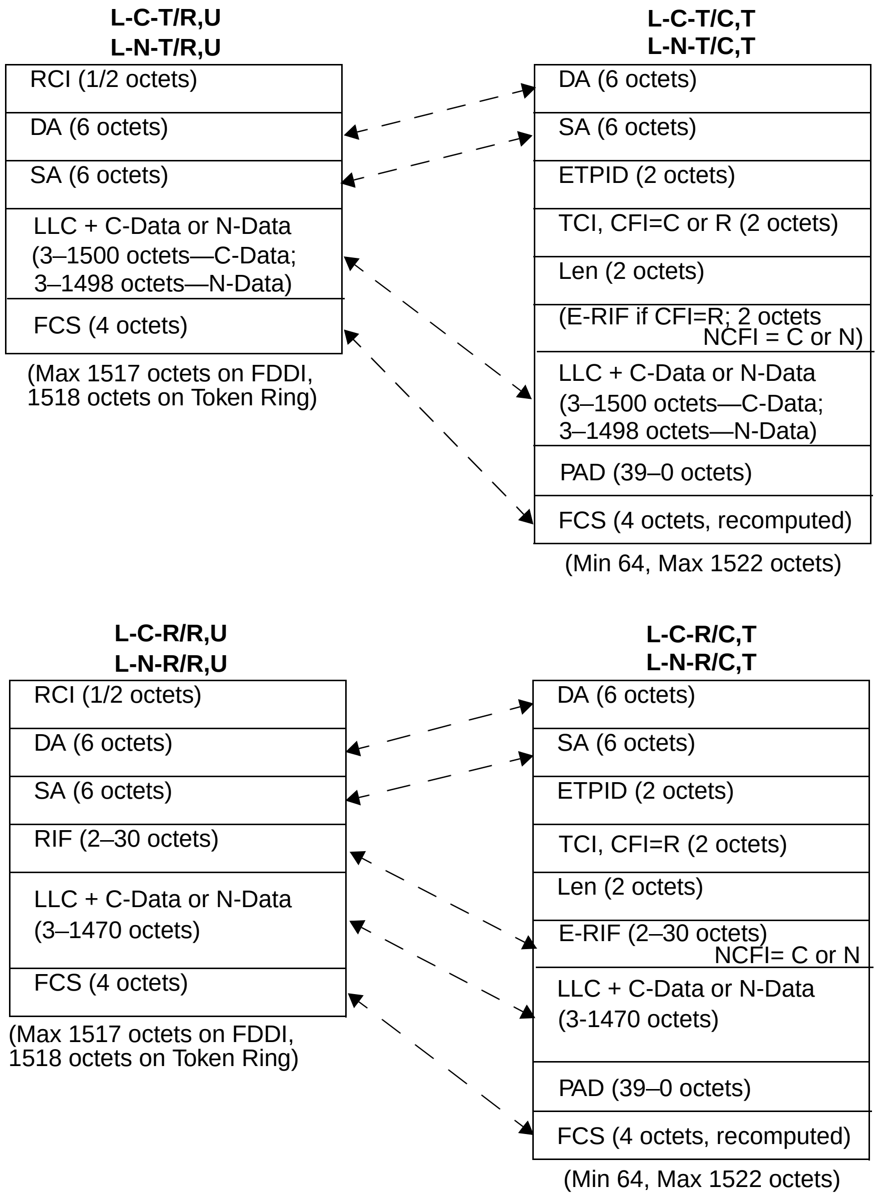


Figure C-13—Translation between L-X-X/R,U and L-X-X/C,T

This form of tagging causes the original frame size to be increased by 5 or 6 octets.

Figure C-14 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-C-T/R,U, L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 8802-5 Token Ring (L-C-T/R,T, L-N-T/R,T, L-C-R/R,T, or L-N-R/R,T). Figure C-14 also illustrates the translation of L-C-T/R,U, L-C-R/R,U, and L-N-R/R,U to tagged frames on FDDI media, the latter two translations illustrating the source-routed form of the tagged frame on FDDI.

Tagging requires the following frame translations:

j) Copy the RCI field;
k) The DA and SA fields carry the same MAC Addresses as in the original frame;

l) Copy the RIF field if present;
m) Insert STPID and TCI, setting the CFI to N or C appropriately;
n) Copy the LLC field;
o) Copy the Data field;
p) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 2—When removing the tag, if the CFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 octets for FDDI and Token Ring.

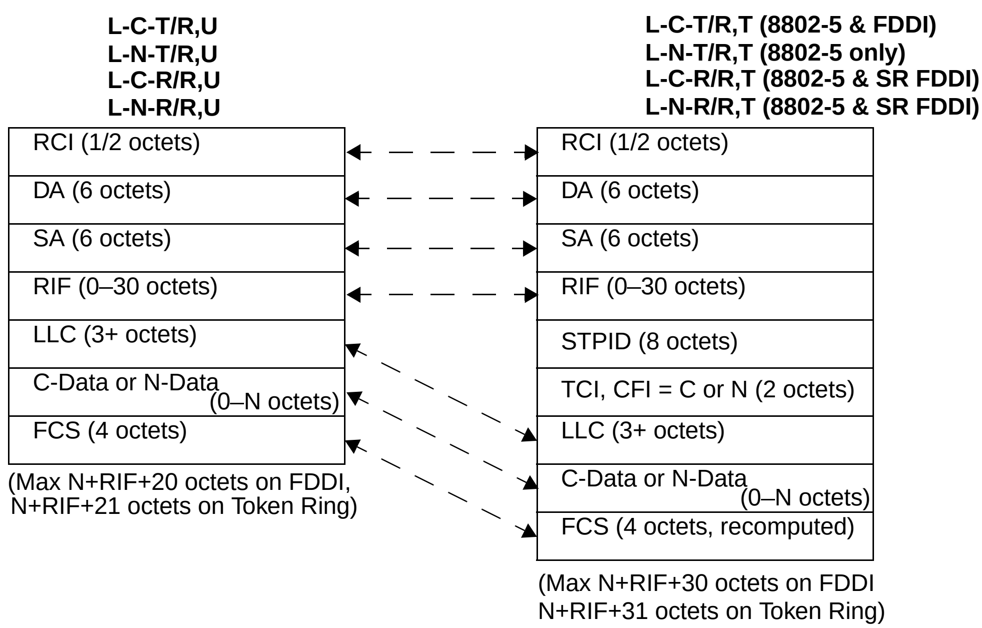


Figure C-14—Translation between L-X-X/R,U and L-X-X/R,T (8802-5 & SR FDDI)

Figure C-15 illustrates the translation between an untagged LLC-encoded frame on Token Ring/FDDI (L-N-T/R,U, L-C-R/R,U, or L-N-R/R,U) and a tagged frame on 8802-5 Token Ring (L-C-T/R,T, L-N-T/R,T, L-C-R/R,T, or L-N-R/R,T). Note that the translation of L-C-T/R,U to L-C-T/R,T was dealt with in Figure C-14.

Tagging requires the following frame translations:

q) Copy the RCI field;
r) The DA and SA fields carry the same MAC Addresses as in the original frame, but with RII reset regardless of its state in the original frame;
s) Insert STPID and TCI, setting the CFI to R;
t) Translate the RIF field if present (RII set in source frame) to the E-RIF, otherwise create an E-RIF with RT indicating a transparent frame. Set the NCFI to C or N appropriately;
u) Copy the LLC field;
v) Copy the Data field;
w) Recompute the FCS.

Removal of the tag (tagged frame to native Token Ring/FDDI frame) involves the reverse of this process.

NOTE 3—When removing the tag, if the NCFI indicates that embedded address information is in a form inappropriate for the destination MAC method, then it is necessary either to translate the address information or to discard the frame.

This form of tagging causes the original frame size to be increased by 10 or 12 octets.

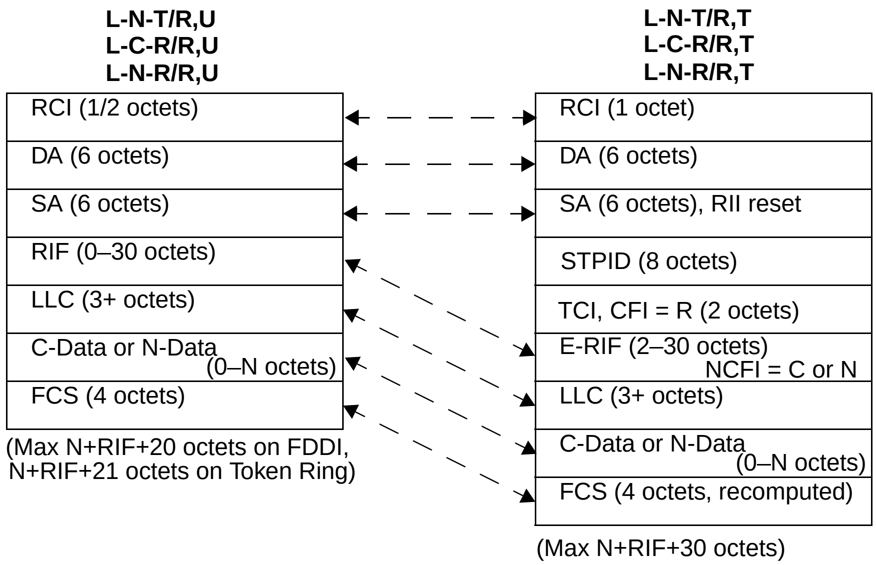


Figure C-15—Translation between L-X-X/R,U and L-X-X/R,T (transparent FDDI)

C.5.3 Translation of tagged frames during relaying

The following subclauses show the frame translations that can occur when a tagged frame is relayed from 802.3/Ethernet to Token Ring/FDDI and vice versa. The translations that occur between the transparent FDDI tagged frame format and the SR form on Token Ring/FDDI are also shown.

C.5.3.1 Tagged frames carrying Ethernet Type-encoded information

Figure C-16 illustrates the translation of tagged frames carrying Ethernet Type-encoded information between Token Ring/FDDI LANs and 802.3/Ethernet LANs.

Relaying Ethernet Type-encoded tagged frames from Token Ring/FDDI (SR form) to 802.3/Ethernet requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Replace the STPID with an ETPID;
d) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). For E-C-T/C,T, CFI = C, otherwise CFI = R;
e) Convert the SPT field to a PT (ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 translation);
f) Copy the RIF, if present, into the tag header E-RIF. Create an E-RIF if the data type being carried is E-N-T/C,T. Set the NCFI in the E-RIF to C or N appropriately;
g) Copy the Data field;
h) Recompute the FCS.

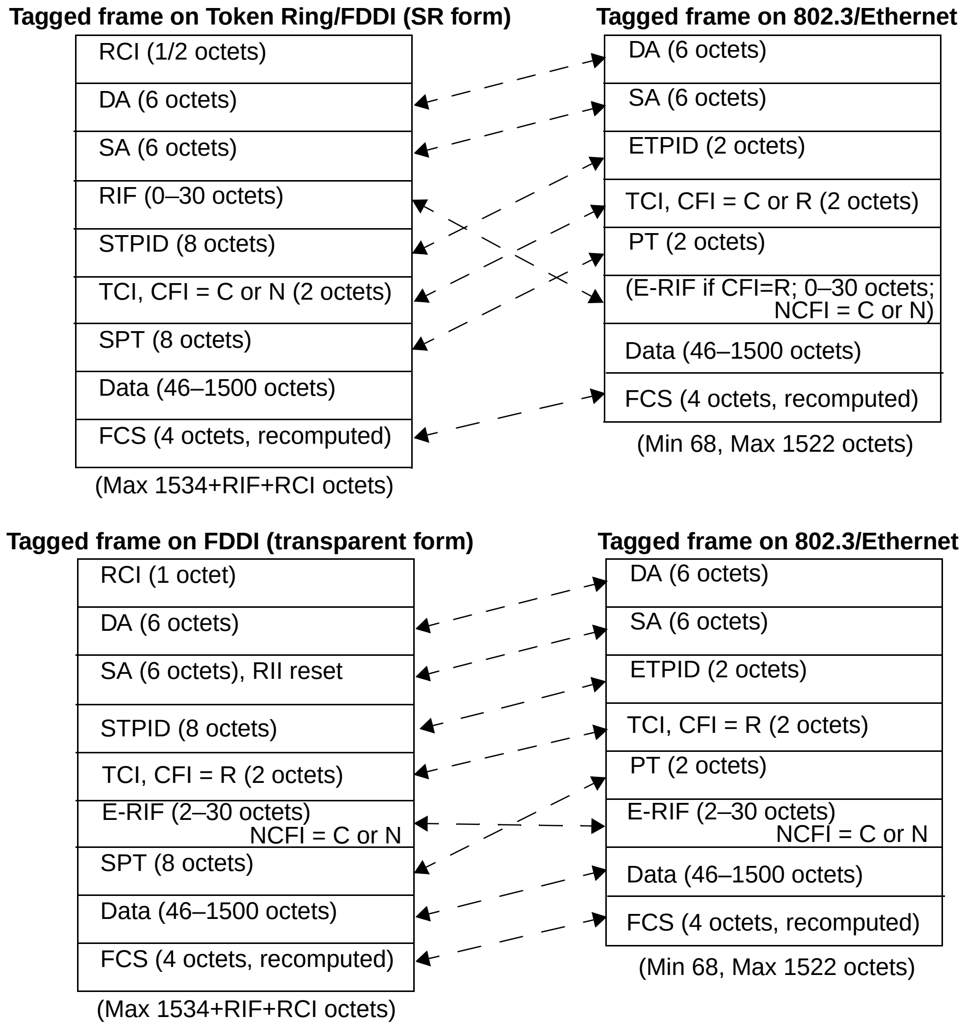


Figure C-16—Relaying Ethernet Type-encoded tagged frames

Relaying Ethernet Type-encoded tagged frames from FDDI (transparent form) to 802.3/Ethernet requires the following frame translations:

i) Remove the RCI field;
j) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
k) Replace the STPID with an ETPID;
l) The TCI field carries the same VID, Priority and CFI values as in the original frame (unless the relay function causes changes to user_priority or VID values);
m) Convert the SPT field to a PT (ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 translation);
n) Copy the E-RIF;
o) Copy the Data field;
p) Recompute the FCS.

Relaying from 802.3/Ethernet to Token Ring/FDDI involves the reverse of these processes.

C.5.3.2 Tagged frames carrying LLC-encoded information

Figure C-17 illustrates the translation of tagged frames carrying LLC-encoded information between Token Ring/FDDI LANs and 802.3/Ethernet LANs.

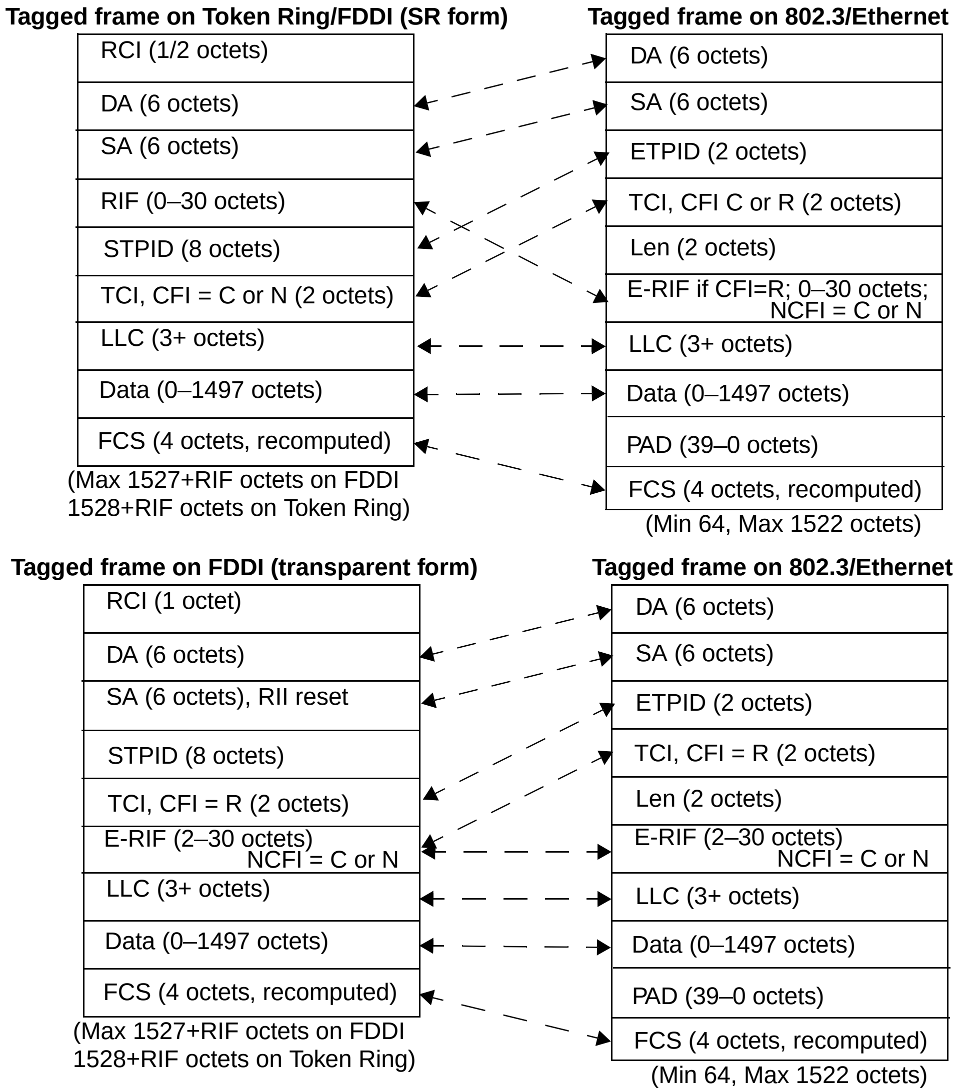


Figure C-17—Relaying LLC-encoded tagged frames

Relaying LLC-encoded frames in tagged format from Token Ring/FDDI (SR form) to 802.3/Ethernet requires the following frame translations:

a) Remove the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
c) Replace the STPID with an ETPID;
d) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). For L-C-T/C,T, the CFI = C, otherwise CFI = R;
e) Insert a LEN field, equal to LLC+RIF (if present) +Data;

f) Copy the RIF, if present, into the tag header E-RIF. Create an E-RIF if the data type being carried is E-N-T/C,T. Set the NCFI bit to C or N appropriately;
g) Copy the LLC field;
h) Copy the Data field;
i) Recompute the FCS.

Relaying LLC-encoded frames in tagged format from FDDI (transparent form) to 802.3/Ethernet requires the following frame translations:

j) Remove the RCI field;
k) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit reset;
l) Replace the STPID with an ETPID;
m) The TCI field carries the same VID, Priority and CFI values as in the original frame (unless the relay function causes changes to user_priority or VID values);
n) Insert a LEN field, equal to E-RIF + LLC +Data;
o) Copy the E-RIF;
p) Copy the LLC field;
q) Copy the Data field;
r) Recompute the FCS.

Relaying from 802.3/Ethernet to Token Ring/FDDI involves the reverse of these process.

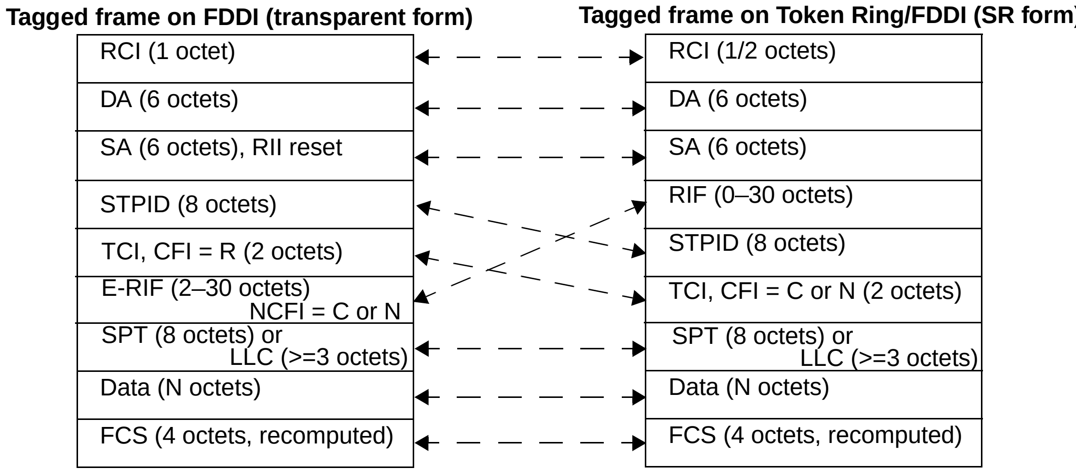
C.5.3.3 Translation between transparent FDDI format and SR format

Figure C-18 illustrates the translation of tagged frames between transparent FDDI format and the corresponding SR format on Token Ring/FDDI LANs. The translation shown applies to X-N-T/R,T, X-C-R/R,T and X-N-R/R,T frames only; other than translation of the RCI field, there is no translation required for X-C-T/R,T frames between Token Ring/FDDI LANs.

Relaying tagged frames from FDDI (transparent form) to Token Ring/FDDI (SR form) requires the following frame translations:

a) Translate the RCI field;
b) The DA and SA fields carry the same MAC Addresses as in the original frame, with the RII bit set or reset to reflect the presence or absence of source-routing information in the E-RIF;
c) Translate source-routing information, if any, from the E-RIF form to the RIF, with the NCFI bit reset;
d) Copy the STPID;
e) The TCI field carries the same VID and Priority values as in the original frame (unless the relay function causes changes to user_priority or VID values). The CFI is set to C or N to match the NCFI value in the E-RIF;
f) Copy the SPT or LLC field;
g) Copy the Data field;
h) Recompute the FCS.

Relaying from Token Ring/FDDI (SR form) to FDDI (transparent form) involves the reverse of these processes.



NOTE—Applies to X-N-T/R,T, X-C-R/R,T, and X-N-R/R,T frames only.

Figure C-18—Relaying tagged frames between transparent and SR forms

C.6 Field definitions

Subclauses C.6.1 through C.6.5 describe the field structures that correspond to some of the field names that appear in abbreviated form in the frame format diagrams in this standard.

NOTE—These fields are defined in other standards, and are not part of the additional specification required for the tagged frame format. They are included here in order to simplify the frame descriptions that appear in this standard, not in order to redefine their structure.

C.6.1 SNAP-encoded Protocol Type

The SNAP-encoded Protocol Type is eight octets in length, encoded in SNAP format. It consists of the standard SNAP header in the first three octets, followed by a SNAP PID consisting of the 00-00-00 OUI, followed by the Ethernet Type value to be encoded, as shown in Figure C-19.

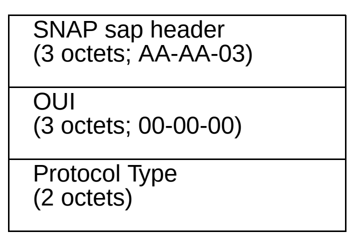


Figure C-19—SNAP-encoded Protocol Type format

C.6.2 Len

This is the IEEE Std 802.3 Length/Type field; for the Length interpretation, it may take any value that is less than or equal to 1500. Values that exceed 1535 are interpreted as Ethernet Types. Values that exceed 1500 but are less than 1535 are undefined.

C.6.3 C-Data and N-Data

This is the data field of the encapsulated frame:

a) N-Data refers to a data field that is carried in Canonical format regardless of the MAC method carrying the frame;
b) C-Data refers to a data field that is carried in Non-canonical format regardless of the MAC method carrying the frame.

C.6.4 RIF and E-RIF

The RIF is the Source-Routing Information Field, as defined in ISO/IEC 15802-3, C.3.3.2. If the original (untagged) frame had a RIF, then the RIF field of the tagged frame takes its value.

The E-RIF is a modified form of the RIF that appears within the tag header in tagged frames on transparent LANs (802.3/Ethernet, and FDDI when used as a transparent LAN). The structure of the E-RIF is defined in 9.3.3.

C.6.5 PAD

Zero or more padding octets, as required in order for the minimum frame size to be at least 64 octets.

Annex D

(informative)

Background to VLANs

The term VLAN has many different definitions throughout the communications industry. The model of VLANs defined in this standard supports most of these views, although it may not be immediately obvious as to how they are supported. The goal of this annex it to take some of the common terms that have been used in the description of VLANs and relate them to the model presented in this standard.

D.1 Basic VLAN concepts

Figure D-1 shows a simple example of a Port-based VLAN. For untagged traffic, VLAN membership for a Port-based VLAN is determined by the PVID assigned to the receiving Port.

NOTE—Other criteria for VLAN membership, such as protocol type or MAC Address, could be used, but these are beyond the scope of this discussion.

For this configuration there needs to be a way to convey the VLAN information between the two bridges. This is done by adding a VLAN tag to every frame that is sent between the two bridges; such frames are known as VLAN-tagged frames. This connection between the two bridges is commonly known as a *Trunk Link*.

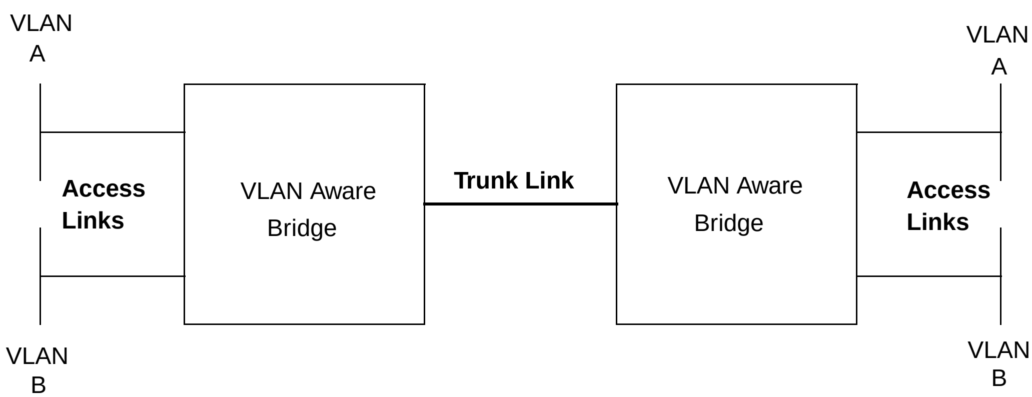


Figure D-1—Port-based VLANs

D.1.1 Trunk Links

A Trunk Link is a LAN segment used for multiplexing VLANs between VLAN Bridges. All the devices that connect to a Trunk Link must be *VLAN-aware*. VLAN-aware devices are devices that are able to understand VLAN membership and VLAN frame formats. Conversely, *VLAN-unaware* devices do not have an understanding of VLAN membership and VLAN frame formats. All frames, including end station frames, on a Trunk Link are VLAN-tagged, i.e., they carry a tag header that contains a non-null VLAN ID. Consequently, there are no VLAN-unaware end stations on an a Trunk Link. The Trunk Link in Figure D-1 is a point-to-point LAN segment; there are therefore exactly two VLAN-aware Bridges attached to this Trunk. A Trunk Link could also be a shared medium LAN segment that has many VLAN-aware Bridges attached to it.

D.1.2 Access Links

The other links in Figure D-1 are commonly known as *Access Links*. An Access Link is a LAN segment used to multiplex one or more VLAN-unaware devices into a Port of a VLAN Bridge. In simple terms this is an 802 LAN segment (IEEE Std 802.3, ISO/IEC 8802-5, etc.) with end stations attached, that is connected into a VLAN-aware Bridge. All frames on an Access Link carry no VLAN identification; i.e., there are no VLAN-tagged frames on an Access Link. Typically the Access Link is viewed as being on the edge of the VLAN network. The Access Link itself could consist of a number of LAN segments interconnected by ISO/IEC 15802-3-conformant Bridges (this is termed a *legacy region* in E.1.2). Like the Access Link, there are no VLAN-tagged frames transmitted in this legacy region.

D.1.3 Hybrid Links

When VLAN-unaware end stations are added to a Trunk Link, the resultant link is commonly known as a Hybrid Link. A Hybrid Link is a LAN segment that has both VLAN-aware and VLAN-unaware devices attached to it. Consequently, a Hybrid Link can carry both VLAN-tagged frames and other (untagged or priority-tagged) frames. It must be borne in mind that, for a given VLAN, all frames transmitted by a given Bridge on a given hybrid link must be tagged the same way on that link. They must be either

a) All untagged; or
b) All tagged, carrying the same VLAN ID.

Note that a Bridge can transmit a mix of VLAN-tagged frames and untagged frames but they must be for different VLANs. In Figure D-2 all the frames for VLANs A and B are tagged on the hybrid link. All frames for VLAN C on the hybrid link are untagged.

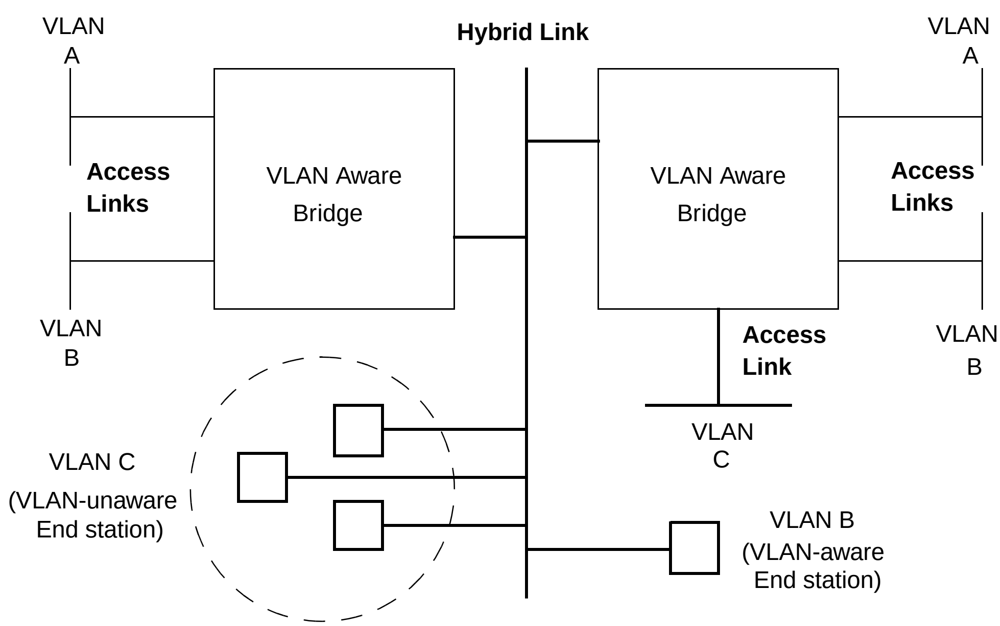


Figure D-2—Hybrid Links

On a Hybrid link the decision to tag or not to tag a frame is a function of the VLAN and not a function of the link itself, since both formats are allowed. The Hybrid link can be thought of as the general case of both Access and Trunk links.

D.2 Relationship with the Port-based VLAN model

D.2.1 Link types

A Hybrid link in which all frames are VLAN-tagged is a Trunk Link. Conversely, a Hybrid Link that has no VLAN-tagged frames is an Access Link. The distinction between Access and Trunk becomes less important in an actual implementation where all types of frames have to be handled. The Acceptable frame Types parameter (8.4.3) allows control to be exerted over the reception of frames that do not carry VLAN identification information (i.e., untagged and priority-tagged frames). Depending upon the value of the parameter, the decision is taken as to whether to discard the frame or to add in VLAN information. The more general implementation will allow all types of frames (VLAN-tagged, priority-tagged, and untagged) to be present, hence all links are conceptually Hybrid Links.

The VLAN model defined in this standard essentially takes the view that all links are Hybrid Links. It is then up to the system administrator to determine, through appropriate application of the management functions available in the Bridge, whether all links remain operating as Hybrid Links, or whether particular links need to be configured as Access Links or Trunk Links, in other words, to explicitly configure some Ports to discard untagged and priority-tagged frames. This allows the description of the functionality of the Bridge to be kept relatively simple, while retaining the practical implementation and configuration mechanisms necessary in order for the other link types to be derived. The Acceptable frame Types parameter does not allow configuration of a Port such that VLAN-tagged frames are discarded; hence, from the practical point of view, the distinction between an Access Link and a Hybrid Link is not one of Port configuration; it is simply determined by the presence or absence of other devices on that link that generate VLAN-tagged frames.

D.2.2 Use of other VLAN styles

This standard defines a Port-based tagging rule, whereby all untagged and priority-tagged frames received by a Port are classified as belonging to the VLAN whose VID (the PVID) is associated with that Port. This Port-based style of operation should be viewed as the base level of a possible hierarchy of VLAN styles, each one able to classify untagged frames according to particular ingress rules. Examples of such ingress rules might include

a) MAC Address-based classification; e.g., associating a set of MAC Addresses with a given VLAN ID in order to define the membership of the VLAN;
b) Protocol-based classification; e.g., allocating VLAN membership on the basis of the higher-layer protocol information carried in the frame;
c) Subnet-based classification; e.g., allocating VLAN membership on the basis of IP subnet addressing characteristics of frames.

For a given implementation, such rules might form a natural hierarchy; e.g., using the above set, IP Subnet-based tagging might take the highest priority. If the packet was not an IP packet, then tagging is based on the protocol being used: IPX or LAT. If some other protocol is in use (not IP, IPX, or LAT), then the classification is based on MAC Addresses. If the addresses in the frame do not match the address-based classifications that are configured, then the Port-based rule is applied.

The result of such a hierarchy is that a given ingress rule defines the default that is applied if the higher priority rule fails to classify the frame, with the Port-based rule forming the lowest level, “catch-all” default.

NOTE—Clearly, if a given rule in the hierarchy is able to classify all possible frames, then all rules below that point in the hierarchy are effectively disabled.

The addition of further ingress rules in 802.1Q Bridges could be achieved

d) As proprietary extensions to the existing specification;
e) As future standardized extensions.

Given that the starting point for this standard is that all links are Hybrid Links, there is no need for such additional classification and tagging functionality to exist within the Bridges themselves; it would, for example, be possible to develop “tagging engines” that are capable of implementing more complex classifications than Port-based classification, and which are placed between the 802.1Q Bridge Port and an Access Link. Such a device would provide a richer functionality in terms of VLAN classification style, while remaining compatible with Port-based VLAN operation.

Annex E

(informative)

Interoperability considerations

VLAN-aware Bridges that conform to this standard are able to interoperate in Bridged LANs with other VLAN-aware Bridges. However, the VLAN-based filtering service defined in this standard, as provided in the context of a single spanning tree for the Bridged LAN, involves some constraints on the network topology and individual device configurations that differ from the set of constraints that apply to the building and configuration of Bridged LANs based only on ISO/IEC 15802-3.

In addition, VLAN-aware Bridges are able to interoperate with Bridges conformant with the ISO/IEC 15802-3 specification (or with the earlier ISO/IEC 10038 specification), as well as with both priority-aware and VLAN-aware end systems. Both the VLAN based filtering service and the tag insertion and removal service of 802.1Q cause constraints on intermixed network topologies and device configurations that again differ from the building and configuration of ISO/IEC 15802-3 standard networks.

The implications of certain device configurations may not be immediately apparent from the technical detail of this standard. In order to clarify the nature of the additional constraints, the following subclauses

a) Describe the basic requirements for interoperability;
b) Discuss those requirements in the context of homogeneous and heterogeneous configurations, with examples of some of the problems that can occur if these requirements are not adhered to.

E.1 Requirements for interoperability

There are two primary aspects of the configuration of a Bridged LAN that are of concern from the point of view of interoperability:

a) Establishing a consistent view of the static filtering configuration of Bridges in the Bridged LAN;
b) Ensuring that untagged frames are VLAN-tagged (and that the tag is subsequently removed) consistently regardless of Spanning Tree reconfigurations.

E.1.1 Static filtering requirements

Static filtering controls allow the network administrator to impose a level of control over the permitted connectivity in the Bridged LAN, by setting static MAC Address filters in the Filtering Databases of Bridges, and by controlling the extent of particular VLANs by manipulation of Static VLAN Registration Entries (8.11.2).

In order to ensure that end station to end station connectivity (or the lack of it) is consistent in all possible Spanning Tree configurations, any static filters need to be established taking account of the full mesh topology of the physical interconnections between Bridges in the Bridged LAN, not just the “normal” Spanning Tree topology to which the network is configured when all Bridges and LAN segments are operating correctly. An example of the consequences of failure to establish consistent controls for static VLAN filtering is given in E.2.1.

E.1.2 Configuration requirements for VLAN-tagging

802.1Q Bridges classify incoming untagged frames by applying a Port-based tagging rule on ingress that uses the PVID for the receiving Port as the VLAN classification for such frames. Maintaining consistent connectivity between any pair of end stations that are on the same VLAN, and where one or both of those end stations is VLAN-unaware, requires that

a) All VLAN-aware Bridge Ports that are connected to the same LAN segment apply a consistent set of ingress rules (8.6);
b) All VLAN-aware Bridge Ports that are connected to the same *legacy region* of a Bridged LAN apply a consistent set of ingress rules;
c) All VLAN-aware Bridge Ports that serve LAN segments to which members of the same VLAN are (or can be) attached apply a consistent set of ingress rules.

A legacy region of a Bridged LAN consists of any set of LAN segments that are physically interconnected via VLAN-unaware, ISO/IEC 15802-3 Bridges. A legacy region has the property that, by appropriate configuration of the Spanning Tree, a Spanning Tree path could be created between any pair of LAN segments in the region such that the path would pass only through VLAN-unaware Bridges.

NOTE—In case b), Spanning Tree reconfiguration within the legacy region can change the logical connectivity between the VLAN Ports and the LAN segments that they (directly or indirectly) serve. Hence, a Spanning Tree reconfiguration could result in any end stations connected to the legacy region being serviced via any of the VLAN-aware Ports. In effect, such a reconfiguration reduces case b) to case a). Figure E-2 and Figure E-3 give examples of this type of configuration. In Figure E-2, the legacy region consists of all three LAN segments and both ISO/IEC 15802-3 Bridges. In Figure E-3, the legacy region consists of the ISO/IEC 15802-3 Bridge and both LAN segments to which it is attached.
An example of case c) is where an end station attached to a leaf LAN segment is in the same VLAN as a server that is attached to a distinct LAN segment, i.e., all possible Spanning Tree paths between the two stations pass through a VLAN-aware region of the Bridged LAN.

The essence of what these rules express is that if a given untagged frame belongs on a given VLAN, then the tagging behavior of any VLAN-aware Bridges that are required to tag that frame needs to be the same, regardless of the logical connectivity that is created by the Spanning Tree configuration of the Bridged LAN. Examples of the consequences of failure to apply these rules appear in E.3 and E.6.

E.2 Homogenous 802.1Q Bridged LANs

This standard requires new considerations in building a Bridged LAN in which all Bridges are VLAN-aware. The arbitrary plug and play capability of ISO/IEC 15802-3 in creating a network topology is restricted when making use of the VLAN extensions defined in this standard.

E.2.1 Consistency of static VLAN filtering

In order for stations that are members of a given VLAN to be able to reach other members of the same VLAN elsewhere in the Bridged LAN, all Ports that are part of the Spanning Tree active topology (i.e., all Ports that are in a forwarding state) connecting the stations must be included in the Member Set (8.11.9) for the given VLAN. In order for this connectivity to be independent of any reconfiguration of the Spanning Tree topology, all paths among those stations, both forwarding and blocked, must have this characteristic. Use of management controls to manipulate the Member Set (e.g., filters for security) must be applied in a manner consistent with requirements of the full mesh topology of the Bridged LAN.

An inconsistency occurs, for example, if a VLAN is restricted from an active path, but not from a redundant path currently blocked by the operation of Spanning Tree. Should a Spanning Tree reconfiguration enable the previously blocked path, the restriction will no longer be in place. In the reverse, a Spanning Tree recon-

figuration may suddenly impose a restriction that had not previously existed. A common use of such management restriction will likely arise from managers who make use of an “access” port construct. An access port may be a port which is absent from the Member Set (8.11.9) in all VLANs but the untagged, default VLAN. Should such an access port become the active connection between two portions of the Bridged LAN as a result of a Spanning Tree reconfiguration, all VLANs but that one will be partitioned at that point in the topology.

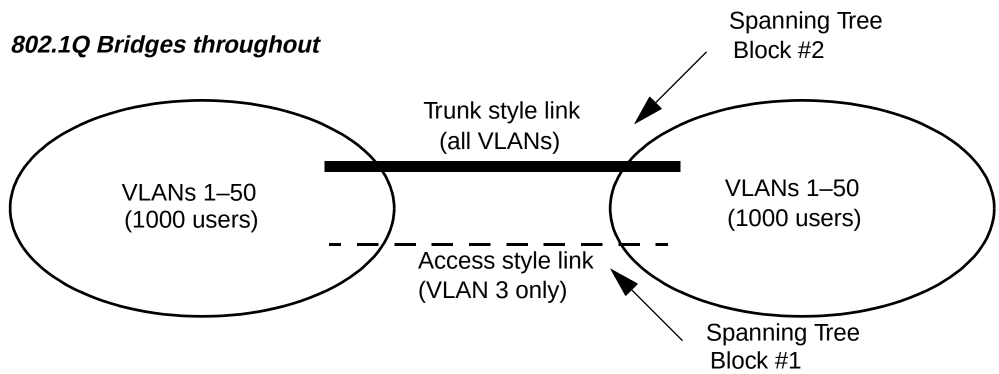


Figure E-1—Static filtering inconsistency

In Figure E-1, the trunk style link and access style link cause a loop through the left and right portions of the network. STP will block one or the other. Should the Spanning Tree block at point #1, all 2000 users may communicate on any of the 50 VLANs. However, should the Spanning Tree block at point #2, the left and right portions of the network will be partitioned on all VLANs excepting VLAN 3 (the VLAN carried via the access style link.)

E.2.2 Consistent view of the “untagged VLAN” on a given LAN segment

In the Port-based VLAN model defined in this standard, the PVID for a Port provides the VLAN classification for all frames on the attached LAN segment that are received in untagged form. Any LAN segment with more than one 802.1Q Bridge attached has such an “untagged VLAN” for each Bridge. No explicit requirement that these be consistent for all Bridges on the same LAN segment, nor mechanism to assure such, has been included in this standard.

Consider the case of a LAN segment to which are attached three VLAN-aware Bridges, each of which is capable of transmission of untagged frames onto the LAN segment. An untagged frame placed on that segment by any one of the Bridges will be associated by each of the other two Bridges with their own configured PVID for their receiving port on that LAN. The 802.1Q VLAN model requires that each frame have a unique VLAN association, and that association is represented by a single, global VID value. Therefore, it follows that all 802.1Q Bridges on that LAN segment must make use of the same PVID for their ports connected to that LAN segment.

It has been suggested that in the special case of a direct point-to-point connection between two 802.1Q Bridges or other VLAN-aware devices, other rules might apply. No mechanism for identifying such links has yet been suggested.

This creates a configuration challenge for installers of Bridges that conform to this standard. Initial management configuration of the Bridges (the setting of PVIDs) must be made consistent among the Bridges, in a manner that takes into account the actual physical topology. Changes to the physical topology may require

specific changes to the configuration of all affected switches. These requirements effectively disallow a plug-and-play installation as supported by ISO/IEC 15802-3 Bridged LANs, unless all Bridges are left with their default PVID configuration of PVID = 1.

E.3 Heterogeneous Bridged LANs: intermixing ISO/IEC 15802-3 (D) and 802.1Q (Q) Bridges

This clause discusses networks in which VLAN-aware Bridges that conform to this standard are intermixed with VLAN-unaware Bridges conformant to the ISO/IEC 15802-3 standard.

A principal limitation in intermixing Q Bridges with D Bridges is that the VLAN filtering services are not universally available throughout the Bridged LAN. Also, services for the insertion and removal of tags are not universally available. Further, spanning tree reconfigurations may cause filtering services, as well as tag insertion and removal services, to become available or become unavailable independent of actions of affected users.

E.3.1 Example: Adding an 802.1Q Bridge to provide filtering to an ISO/IEC 15802-3 network

Example problems can be shown with the following topology diagrams. Figure E-2 includes one Q Bridge and two D Bridges:

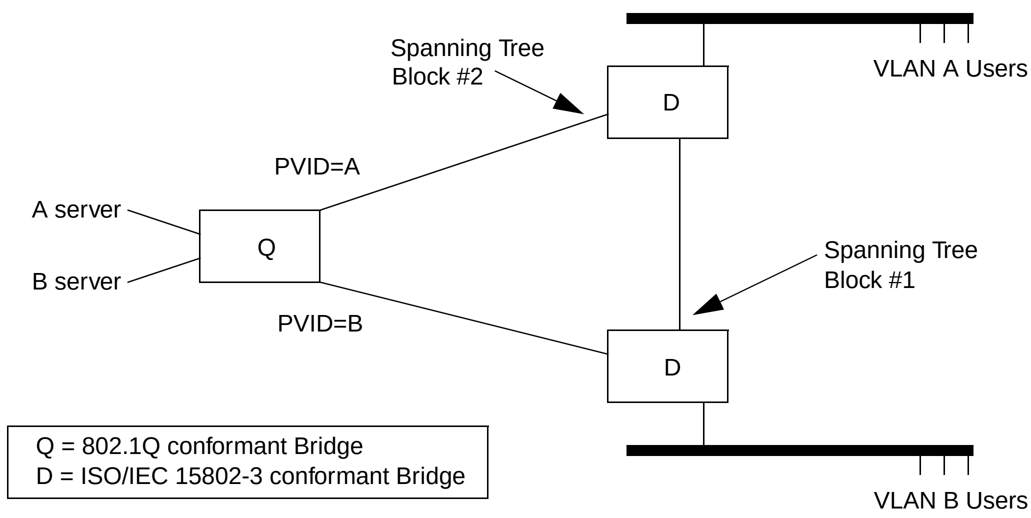


Figure E-2—Interoperability with ISO/IEC 15802-3 Bridges: example 1

If the Spanning Tree protocol determines to break the loop among the three Bridges by blocking at point #1, connectivity within each VLAN is as desired. However, should the block occur at point #2, traffic from VLAN A users will pass through both D Bridges, and be treated as VLAN B traffic upon arrival in the Q Bridge. Connectivity to the A server will be lost for the A users.

E.3.2 Example: Adding an ISO/IEC 15802-3 Bridge to a (previously) Homogenous 802.1Q Network

A similar problem, demonstrating the impact of placing a D Bridge within an otherwise homogenous Q topology, can be shown by the configuration in Figure E-3. Here we include two Q Bridges and add a single redundant D Bridge:

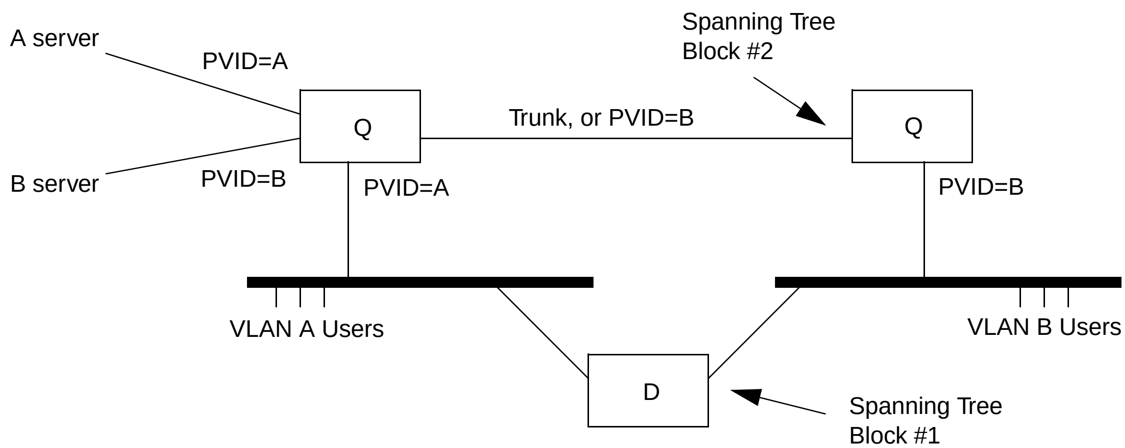


Figure E-3—Interoperability with ISO/IEC 15802-3 Bridges: example 2

If STP determines to break the loop among the three Bridges by blocking at point #1, connectivity within each VLAN is as desired. The two Q switches operate as expected. A and B VLAN frames are VLAN-tagged on arrival in either Q Bridge, and forwarded only to the appropriate servers. Now suppose an STP reconfiguration results in a block at point #2, but not at #1. This redirects VLAN B user traffic through the ISO/IEC 15802-3 Bridge. VLAN B users no longer have their traffic identifiably distinct from VLAN A. An immediate consequence is that the VLAN B users will no longer have access to the “B server.”

E.4 Heterogeneous Bridged LANs: intermixing ISO/IEC 11802-5 and 802.1Q Bridges

Translating Bridges (i.e., Bridges that relay between Token Ring/FDDI and 802.3/Ethernet LANs) that implement the encapsulation techniques described in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390 can be intermixed with 802.1Q Bridges under certain limited conditions. In order to understand the limitations involved, it is necessary to describe what happens to the various tagged frame formats when passed through a Translating Bridge. The frame formats shown in the following subclauses use the notation that is defined in Annex C.

NOTE—The examples shown are not exhaustive; in particular, the examples do not make use of the transparent tagged frame format on FDDI. However, the examples illustrate the nature of the problems that can occur with such translations.

E.4.1 LLC-encoded tagged frames relayed from 802.3/Ethernet to Token Ring or source-routed FDDI

A Transparent LLC-encoded frame on 802.3/Ethernet has the following form:

L-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

The Translating Bridge will recognize the ETPID as an Ethernet Type that requires translation into an SPT. This effectively translates the ETPID to an STPID. The translated frame therefore appears as follows:

RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LLC, C-Data, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, it includes spurious LEN and PAD fields, and is therefore not a valid tagged frame.

A source-routed LLC-encoded frame on 802.3/Ethernet has the following form:

L-C-R/C,T: DA, SA, ETPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

The Translating Bridge generates

RCI, DA, SA (RII reset), STPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), LLC, C-Data, FCS

Again, the resultant frame superficially appears to be a valid tagged frame; however, it includes spurious LEN, RIF and PAD fields, the RIF information is not visible to any source routing Bridges attached to the Ring medium, and the CFI indicates Non-canonical data where the actual data is Canonical.

NOTE—Similar problems exist for the other two LLC-encoded formats, L-N-R and L-N-T, but the CFI correctly indicates Non-canonical data in the translated frame.

In both cases, the effect of the Translating Bridge is symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) In both cases, any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;
d) In the case of the source-routed frame, any source routing Bridges attached to the Ring will treat the frame as a transparent frame;
e) As the effect of the Translating Bridge is symmetric, passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.2 Ethernet Type-encoded tagged frames relayed from 802.3/Ethernet to Token Ring or source-routed FDDI

A Transparent Ethernet Type-encoded frame on 802.3/Ethernet has the following form:

E-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), PT, C-Data, FCS

The Translating Bridge generates:

RCI, DA, SA (RII reset), STPID, TCI (CFI reset), PT, C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), SPT, C-Data, FCS

The resultant frame is superficially a valid tagged frame, but carries an untranslated Ethernet Type where there should be a SNAP-encoded Ethernet Type.

A source-routed Ethernet Type-encoded frame on 802.3/Ethernet has the following form:

E-C-R/C,T: DA, SA, ETPID, TCI (CFI set), PT, RIF (NCFI=C), C-Data, FCS

The Translating Bridge generates

RCI, DA, SA, STPID, TCI (CFI set), PT, RIF (NCFI=C), C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), SPT, C-Data, FCS

Again, the resultant frame is superficially a valid tagged frame, but carries an untranslated Ethernet Type where there should be a SNAP-encoded Ethernet Type, and carries a spurious RIF field. The CFI is also incorrect.

NOTE—Similar problems exist for the other two Ethernet Type-encoded formats, E-N-R and E-N-T, but the CFI correctly indicates Non-canonical data in the translated frame.

In both cases, the effect of the Translating Bridge is symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;
d) In the case of the source-routed frame, any source routing Bridges attached to the Ring will treat the frame as a transparent frame;
e) As the effect of the Translating Bridge is symmetric, passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.3 LLC-encoded tagged frames relayed from Token Ring or source-routed FDDI to 802.3/Ethernet

A Transparent LLC-encoded frame on Token Ring/FDDI has the following form:

L-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), LLC, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), LEN, LLC, C-Data, PAD, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, it is missing the LEN field, and is therefore not a valid tagged frame.

A source-routed LLC-encoded frame on Token Ring/FDDI has the following form:

L-C-R/R,T: RCI, DA, SA (RII set), RIF, STPID, TCI (CFI reset), LLC, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), LLC, C-Data, PAD, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

L-C-R/C,T: DA, SA, ETPID, TCI (CFI set), LEN, RIF (NCFI=C), LLC, C-Data, PAD, FCS

Again, the resultant frame superficially appears to be a valid tagged frame, but is missing the LEN field. The RIF information has been lost.

Similar problems exist for the other two LLC-encoded formats, L-N-R and L-N-T, but in addition, the CFI will be set, indicating the presence of a RIF in the tag header when no such RIF field exists.

In both cases, the effect of the Translating Bridge is almost symmetrical; passing the invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet, but with the loss of any source-routing information that may have been present, and the inclusion of a PAD field if the original frame on the Ring medium had been small.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag the translated frames will generate invalid untagged frames;
c) Any 802.1Q Bridge that attempts to relay the translated frames back onto Token Ring/FDDI will generate invalid tagged frames;
d) Any source-routing information is lost;
e) As the effect of the Translating Bridge is almost symmetric (the RIF is lost, and there may be a PAD included), passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices, as long as those devices are not sensitive to the presence of spurious PAD information.

E.4.4 Ethernet Type-encoded tagged frames relayed from Token Ring or source-routed FDDI to 802.3/Ethernet

A Transparent Ethernet Type-encoded frame carrying Canonical data on Token Ring/FDDI has the following form:

E-C-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI reset), SPT, C-Data, FCS

The Translating Bridge generates

DA, SA, ETPID, TCI (CFI reset), SPT, C-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in

E-C-T/C,T: DA, SA, ETPID, TCI (CFI reset), PT, C-Data, FCS

As can be seen, the resultant frame superficially appears to be a valid tagged frame; however, the SPT has not undergone translation to a PT. End stations encountering this frame on 802.3/Ethernet would only be capable of interpreting it if they were able to recognize the SPT as an embedded Ethernet Type.

Translating Canonical source-routed Ethernet Type-encoded information produces similar results, but with the additional loss of the source-routing information.

A Transparent Ethernet Type-encoded frame carrying Non-canonical information on Token Ring/FDDI has the following form:

E-N-T/R,T: RCI, DA, SA (RII reset), STPID, TCI (CFI set), SPT, N-Data, FCS

The Translating Bridge generates:

DA, SA, ETPID, TCI (CFI set), SPT, N-Data, FCS

Whereas a true translation of the tagged frame via an 802.1Q Bridge would result in:

E-N-T/C,T: DA, SA, ETPID, TCI (CFI set), PT, RIF (NCFI=N), N-Data, FCS

Again, the resultant frame superficially appears to be a valid tagged frame, but the SPT has not been translated to a PT, and the RIF is not present in the tag header. Any 802.1Q device will therefore interpret the first N octets of the N-Data field as if it was the RIF.

Translating Non-canonical source-routed Ethernet Type-encoded information produces similar results, but with the additional loss of the source-routing information.

In both cases, the effect of the Translating Bridge is almost symmetrical; passing this invalid frame back through the Translating Bridge restores it to a valid tagged frame format on 802.3/Ethernet, but with the loss of any source-routing information that may have been present.

Consequently,

a) The translated frames cannot be correctly interpreted by VLAN-aware end stations attached to the Ring medium;
b) Any 802.1Q Bridge that attempts to untag translated frames carrying Non-canonical information will generate invalid untagged frames;

c) Any 802.1Q Bridge that attempts to relay untag translated frames carrying Non-canonical information back onto Token Ring/FDDI will generate invalid tagged frames;
d) Frames carrying Canonical information can be successfully untagged or relayed in tagged form using other MAC methods, as long as the 802.1Q Bridge is capable of correctly handling the embedded SPT;
e) Any source-routing information is lost;
f) As the effect of the Translating Bridge is symmetric (apart from the loss of RIF), passing the frame through an even number of such translations before any 802.1Q device attempts to interpret the frame format results in correct operation of the 802.1Q devices.

E.4.5 Conclusions

Except for the limited case of relaying Canonical Ethernet Type-encoded information from 802.3/Ethernet to Token Ring/FDDI, the translation that a tagged frame undergoes when passing through a Translating Bridge renders the frame uninterpretable by any 802.1Q device (either end station or Bridge), unless the frame has passed through an even number of Translating Bridges before the 802.1Q device attempts to interpret the frame. In regions where an odd number of translations have occurred, source-routing information is rendered invisible to source routing Bridges in some cases. In the other cases, source-routing information is lost after the first translation.

Consequently, the use of Translating Bridges intermixed with 802.1Q Bridges is feasible only if

a) An even number of translations (or zero translations) is experienced by any tagged frame that is transmitted between any pair of 802.1Q-aware devices in the Bridged LAN;
b) The loss of source routing capability across some regions of the Bridged LAN is acceptable; specifically across regions where the first Translating Bridge encountered by a correctly formatted tagged frame will relay the frame from Token Ring/FDDI to 802.3/Ethernet;
c) End stations are not sensitive to receiving LLC-encoded frames that have PAD octets added to the LLC user data.

E.5 Heterogeneous Bridged LANs: intermixing 802.1Q Bridges with ISO/IEC 15802-3 Bridges

The specification in this standard for the use of GMRP in VLANs (11.2) makes use of VLAN-tagged frames to signal the GIP Context that applies to the registration information carried in GMRP PDUs. Devices that implement GMRP as specified in ISO/IEC 15802-3 will regard such frames as badly formed GMRP frames, and will therefore discard them on receipt. Using an ISO/IEC 15802-3 Bridge to interconnect two or more LAN regions containing 802.1Q devices that implement GMRP will therefore prevent GMRP information propagation between the 802.1Q regions, with attendant effects upon the forwarding behavior of both the ISO/IEC 15802-3 and 802.1Q Bridges in the LAN. This configuration can be made to work if the ISO/IEC 15802-3 Bridge is statically configured with

a) An All Groups entry in the Filtering Database, specifying Registration Fixed on all Ports, and
b) The GMRP Protocol Administrative Control parameters set to disable GMRP on all Ports.

As the Bridge no longer supports the GMRP application, it will forward GMRP PDUs on all Ports that are in Forwarding. The effect of this is to configure the ISO/IEC 15802-3 Bridge to behave in the same manner as an ISO/IEC 10038 Bridge.

Placing ISO/IEC 15802-3 Bridges around the periphery of an 802.1Q-based Bridged LAN works correctly, as long as, for a given ISO/IEC 15802-3 Bridge, the 802.1Q Bridges connected to the same segment(s) are configured to untag any VLANs that are relevant to the GMRP operation of the ISO/IEC 15802-3 Bridge.

The ISO/IEC 15802-3 Bridge generates untagged GMRP frames, which the 802.1Q Bridges classify according to the value of the PVID for the reception Port; in a simple configuration of the 802.1Q Bridges, the Ports that connect to the ISO/IEC 15802-3 Bridge are configured for the PVID VLAN to be untagged on egress.

NOTE—There may be situations where more complex configurations are required, in which VLANs other than the PVID are configured untagged in order to maintain the correct ISO/IEC 15802-3 Bridge filtering behavior.

The effect of this type of configuration is that all registrations propagated by a given ISO/IEC 15802-3 Bridge on a given (Port-based) VLAN are seen by all other ISO/IEC 15802-3 Bridges served by 802.1Q Bridges for which that VLAN is configured for untagged egress. The filtering behavior of the ISO/IEC 15802-3 Bridges is therefore governed only by the behavior of other devices (both ISO/IEC 15802-3 and 802.1Q) that are attached to the same VLAN.

E.6 Intermixing 802.1Q Version 1.0 Bridges with future 802.1Q Bridges

The discussion above on intermixing Q Bridges with D Bridges has a direct analogue in the plan to provide a simple VLAN standard (Q version 1.0) initially, and later to provide extensions to the standard (Q version 2.0 on) which extends the VLAN capabilities to support more sophisticated ingress rules for frame classification. Some of the topology restrictions will probably be similar to the “Q intermixed with D” cases.

E.6.1 Example: Intermixing Layer 3 Ingress Rules

Consider the case where a “Qv2” switch capability is specified allowing for classification of frames by protocol. This would allow support for IP and IPX as distinct VLANs. The following diagram might apply when a Qv2 Bridge is added to a version 1.0 topology to allow users of two protocols to participate in two separate VLANs:

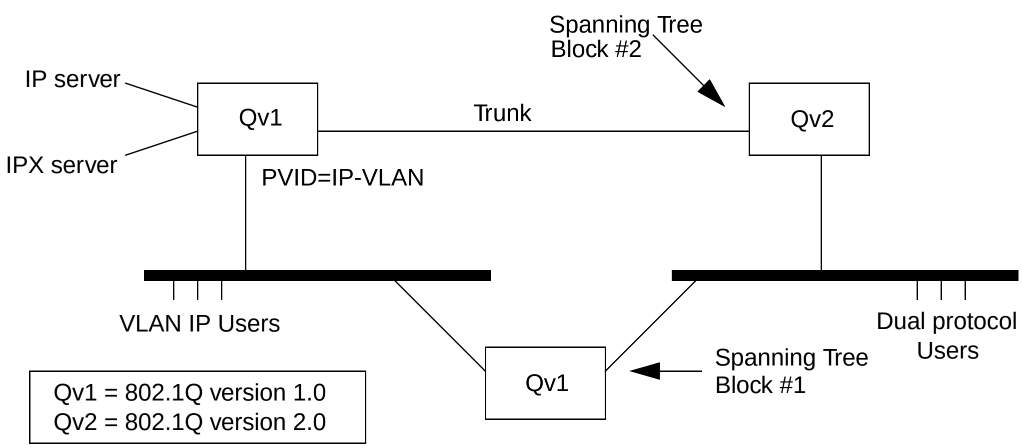


Figure E-4—Interoperability between Q versions 1 and 2

Consider this network, when STP has blocked at point #1, and not at #2. The upper Qv1 switch operates as expected, and the Qv2 switch provides protocol based classification for the frames received from the dual protocol users. IP and IPX VLAN frames are VLAN-tagged on the trunks to and from the uppermost Bridge and servers. But if a STP reconfiguration should result in a block at point #2, but not at #1, activating traffic through the lower Qv1 Bridge, dual protocol users will have all their traffic treated as part of the IP-VLAN. An immediate consequence is that the uppermost Bridge will no longer provide them access to the “IPX server.”

E.6.2 Differing views of untagged traffic on a given LAN segment

Further challenges arise when one considers the case where several Q Bridges, some of version 1 and some of version 2, all attach to the same LAN segment. Again, the rule that any given frame exists in exactly one VLAN requires that all of these Bridges be configured with the same ingress rules. In this case, the Q1 Bridges will provide a least common capability, and further require common configuration of the PVID.

E.6.3 Interoperability with 802.1Q Version 2.0 offering multiple spanning trees

Several very different architectures for multiple spanning tree support have been discussed, but none define an architecture sufficiently for analysis of interoperability with the version 1.0 defined in this standard. The benefits of such an architecture have been discussed, and among these are relaxation of many, but not all, of the restrictions discussed in this subclause. In particular, the multiple spanning tree models appear to offer easier integration within both homogenous environments and in networks intermixing D Bridges with multiple spanning tree VLAN-aware devices.

Annex F

(informative)

Frame translation considerations

When relaying frames between different MAC methods, there are a number of frame translations that may be required, depending upon the source and destination MAC methods concerned, the tagging state of the received and transmitted frames, and the data carried within the frames:

a) If the source and destination MAC methods differ, then the overall format of the received frame must be translated into the frame format required for the MAC onto which the frame will be transmitted. The details of the frame translation at this level is defined by the definition of the Internal Sublayer Service (ISO/IEC 15802-3, 6.4), the support of the internal sublayer service by specific MAC procedures (ISO/IEC 15802-3, 6.5), and the standards that define the specific MAC procedures concerned.
b) If the received frame is being relayed in tagged format between differing MAC methods, or if the tagging state is to change, then the tag header may require translation, insertion, or removal as defined in this standard.
c) If Ethernet Type-encoded data is being carried in the frame, then the format of that data may require translation as defined in ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390. These translations are essentially concerned with providing a standardized means of representing Ethernet Type-encoded frames on LANs that have no inherent ability within their MAC procedures to represent Ethernet Type values (i.e., LAN MACs where the “native” link layer protocol is LLC), and ensuring that the frame translation in the reverse direction results in the correct format on the 802.3/Ethernet LAN. The mechanisms specified in these standards are based around the use of SNAP encoding to carry Ethernet Type values in native LLC-encoded environments, and the use of translation tables to control the reverse translation if this should be necessary.

One aspect that is not fully covered in the standards mentioned is the issue of how differences in bit ordering between MAC methods can affect the data translation requirements. For the majority of data relayed by a Bridge, these differences in bit ordering are not an issue; although the bit ordering of MAC data “on the wire” may differ between MAC methods, the representation of that data within the relay function of the Bridge is independent of the order of transmission or reception adopted by the various MACs.

An exception can occur when the data carried in a frame includes MAC Address values; for example, in IP ARP packets. There are two formats used in LANs when representing MAC Address values in MAC user data:

d) Canonical format;
e) Non-canonical format.

The octet ordering is identical in both cases (the first octet put into the mac_service_data_unit is the left-most octet of the address when written down using hexadecimal notation as defined in Section 5 of IEEE Std 802); however, the bit ordering within each octet differs:

f) Canonical format: The least significant bit of each octet of the standard hexadecimal representation of the address (see IEEE Std 802-1990) represents the least-significant (LS) bit of the corresponding octet of the Canonical format of the address;
g) Non-canonical format: The most significant bit of each octet of the standard hexadecimal representation of the address (see IEEE Std 802-1990) represents the least-significant bit of the corresponding octet of the Canonical format of the address.

Ideally, a single format would be used in all places; however, both can be used, depending on the MAC method concerned. The native format used in MACs where the bit transmission order is LS bit first (e.g., IEEE Std 802.3 and ISO/IEC 8802-4), and also in FDDI, is the Canonical format; in MACs where the bit transmission order is most-significant (MS) bit first (e.g., ISO/IEC 8802-5), Non-canonical format is used. A further complication here is that the format used is not consistent across all higher layer protocols (some use a single format regardless of MAC method, others use the native format for the underlying MAC method), and there is no single standard that specifies which protocols carry embedded MAC Address information or the format in which they carry it.

Clearly, in order for the recipients of frames carrying embedded address information to be able to interpret MAC Address information correctly, it is necessary either to include information in the frame that specifies which format is in use, or for Bridges to modify the format of the information appropriately when relaying frames between regions of the network where the expected format differs for the protocol being carried.

In “Translating Bridges” (Bridges that relay between differing MAC methods, based on ISO/IEC 15802-3, ISO/IEC 11802-5, IETF RFC 1042, and IETF RFC 1390—i.e., in Bridging Function B1 in Figure C-2), the latter approach is the only option, as there is no way that additional information can be carried in the frame in order to identify the embedded format. Consequently, in order for such a Bridge to successfully translate embedded address information, it needs to be able to recognize the higher layer protocol carried in the frame, and act accordingly.

In VLAN-aware Bridges, the information carried in the CFI (and in the NCFI bit of the RIF, in 802.3/Ethernet frames) provides a direct indication of the format of embedded addresses. Hence, any considerations related to translating the format of embedded addresses can be confined to Bridges where a frame is received untagged and the tag header is inserted on transmission, or received tagged and the tag header is removed on transmission.

EXHIBIT 4B

Internet Draft Document
L2VPN Working Group
draft-ietf-l2vpn-vpls-ldp-08.txt
Expires: May 2006

Marc Lasserre
Vach Kompella
(Editors)
November 2005

Virtual Private LAN Services over MPLS

Status of this Memo

IPR Disclosure Acknowledgement

Abstract

This document describes a Virtual Private LAN Service (VPLS) solution using pseudo-wires, a service previously implemented over other tunneling technologies and known as Transparent LAN Services (TLS). A VPLS creates an emulated LAN segment for a given set of users, i.e., it creates a Layer 2 broadcast domain that is fully capable of learning and forwarding on Ethernet MAC addresses that is closed to a given set of users. Multiple VPLS services can be supported from a single PE node.

This document describes the control plane functions of signaling pseudo-wire labels using LDP [RFC3036], extending [PWE3-CTRL]. It is agnostic to discovery protocols. The data plane functions of forwarding are also described, focusing, in particular, on the

learning of MAC addresses. The encapsulation of VPLS packets is described by [PWE3-ETHERNET].

1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

2. Table of Contents

3. Introduction

Ethernet has become the predominant technology for Local Area Network (LAN) connectivity and is gaining acceptance as an access technology, specifically in Metropolitan and Wide Area Networks (MAN and WAN, respectively). The primary motivation behind Virtual Private LAN Services (VPLS) is to provide connectivity between geographically dispersed customer sites across MANs and WANs, as if they were connected using a LAN. The intended application for the end-user can be divided into the following two categories:

- Connectivity between customer routers: LAN routing application
- Connectivity between customer Ethernet switches: LAN switching application

Broadcast and multicast services are available over traditional LANs. Sites that belong to the same broadcast domain and that are connected via an MPLS network expect broadcast, multicast and unicast traffic to be forwarded to the proper location(s). This requires MAC address learning/aging on a per pseudo-wire basis, packet replication across pseudo-wires for multicast/broadcast traffic and for flooding of unknown unicast destination traffic.

[PWE3-ETHERNET] defines how to carry Layer 2 (L2) frames over point-to-point pseudo-wires (PW). This document describes extensions to [PWE3-CTRL] for transporting Ethernet/802.3 and VLAN [802.1Q] traffic across multiple sites that belong to the same L2 broadcast domain or VPLS. Note that the same model can be applied to other 802.1 technologies. It describes a simple and scalable way to offer Virtual LAN services, including the appropriate flooding of broadcast, multicast and unknown unicast destination traffic over MPLS, without the need for address resolution servers or other external servers, as discussed in [L2VPN-REQ].

The following discussion applies to devices that are VPLS capable and have a means of tunneling labeled packets amongst each other. The resulting set of interconnected devices forms a private MPLS VPN.

3.1. Terminology

Q-in-Q 802.1ad Provider Bridge extensions also known

as stackable VLANs or Q-in-Q.

Qualified learning Learning mode in which each customer VLAN is mapped to its own VPLS instance.

Service delimiter Information used to identify a specific customer service instance. This is typically encoded in the encapsulation header of customer frames (e.g. VLAN Id).

Tagged frame Frame with an 802.1Q VLAN identifier.

Unqualified learning Learning mode where all the VLANs of a single customer are mapped to a single VPLS.

Untagged frame Frame without an 802.1Q VLAN identifier

3.2. Acronyms

AC Attachment Circuit

BPDU Bridge Protocol Data Unit

CE Customer Edge device

FEC Forwarding Equivalence Class

FIB Forwarding Information Base

LAN Local Area Network

LDP Label Distribution Protocol

MTU-s Multi-Tenant Unit switch

PE Provider Edge device

PW Pseudo-wire

STP Spanning Tree Protocol

VLAN Virtual LAN

VLAN tag VLAN Identifier

4. Topological Model for VPLS

An interface participating in a VPLS must be able to flood, forward, and filter Ethernet frames. Figure 1 below shows the topological model of a VPLS. The set of PE devices interconnected via PWs appears as a single emulated LAN to customer X. Each PE will form remote MAC address to PW associations and associate

directly attached MAC addresses to local customer facing ports. This is modeled on standard IEEE 802.1 MAC address learning.

```
  +-----+                                                  +-----+
  | CE1 +---+      ...........................     +---| CE2 |
  +-----+   |      .                          .     |   +-----+
   Site 1   |   +----+                     +----+   |   Site 2
            +---| PE |       Cloud         | PE |---+
                +----+                     +----+
                   .                          .
                   .          +----+          .
                   ..........| PE |...........
                              +----+         ^
                                |            |
                                |            +-- Emulated LAN
                              +-----+
                              | CE3 |
                              +-----+
                              Site 3
```

Figure 1: Topological Model of a VPLS for Customer X
With three sites

We note here again that while this document shows specific examples using MPLS transport tunnels, other tunnels that can be used by PWs (as mentioned in [PWE-CTRL]), e.g., GRE, L2TP, IPSEC, etc., can also be used, as long as the originating PE can be identified, since this is used in the MAC learning process.

The scope of the VPLS lies within the PEs in the service provider network, highlighting the fact that apart from customer service delineation, the form of access to a customer site is not relevant to the VPLS [L2VPN-REQ]. In other words, the attachment circuit (AC) connected to the customer could be a physical Ethernet port, a logical (tagged) Ethernet port, an ATM PVC carrying Ethernet frames, etc., or even an Ethernet PW.

The PE is typically an edge router capable of running the LDP signaling protocol and/or routing protocols to set up PWs. In addition, it is capable of setting up transport tunnels to other PEs and delivering traffic over PWs.

4.1. Flooding and Forwarding

One of attributes of an Ethernet service is that frames sent to broadcast addresses and to unknown destination MAC addresses are flooded to all ports. To achieve flooding within the service provider network, all unknown unicast, broadcast and multicast frames are flooded over the corresponding PWs to all PE nodes participating in the VPLS, as well as to all ACs.

Note that multicast frames are a special case and do not necessarily have to be sent to all VPN members. For simplicity,

the default approach of broadcasting multicast frames can be used. The use of IGMP snooping and PIM snooping techniques should be used to improve multicast efficiency. A description of these techniques is beyond the scope of this document.

To forward a frame, a PE MUST be able to associate a destination MAC address with a PW. It is unreasonable and perhaps impossible to require PEs to statically configure an association of every possible destination MAC address with a PW. Therefore, VPLS-capable PEs SHOULD have the capability to dynamically learn MAC addresses on both ACs and PWs and to forward and replicate packets across both ACs and PWs.

4.2. Address Learning

Unlike BGP VPNs [BGP-VPN], reachability information is not advertised and distributed via a control plane. Reachability is obtained by standard learning bridge functions in the data plane.

When a packet arrives on a PW, if the source MAC address is unknown, it needs to be associated with the PW, so that outbound packets to that MAC address can be delivered over the associated PW. Likewise, when a packet arrives on an AC, if the source MAC address is unknown, it needs to be associated with the AC, so that outbound packets to that MAC address can be delivered over the associated AC.

Standard learning, filtering and forwarding actions, as defined in [802.1D-ORIG], [802.1D-REV] and [802.1Q], are required when a PW or AC state changes.

4.3. Tunnel Topology

PE routers are assumed to have the capability to establish transport tunnels. Tunnels are set up between PEs to aggregate traffic. PWs are signaled to demultiplex encapsulated Ethernet frames from multiple VPLS instances that traverse the transport tunnels.

In an Ethernet L2VPN, it becomes the responsibility of the service provider to create the loop free topology. For the sake of simplicity, we define that the topology of a VPLS is a full mesh of PWs.

4.4. Loop free VPLS

If the topology of the VPLS is not restricted to a full mesh, then it may be that for two PEs not directly connected via PWs, they would have to use an intermediary PE to relay packets. This topology would require the use of some loop-breaking protocol, like a spanning tree protocol.

Instead, a full mesh of PWs is established between PEs. Since every PE is now directly connected to every other PE in the VPLS via a PW, there is no longer any need to relay packets, and we can instantiate a simpler loop-breaking rule - the "split horizon" rule: a PE MUST NOT forward traffic from one PW to another in the same VPLS mesh.

Note that customers are allowed to run a Spanning Tree Protocol (STP) (e.g., as defined in [802.1D-REV]), such as when a customer has "back door" links used to provide redundancy in the case of a failure within the VPLS. In such a case, STP Bridge PDUs (BPDUs) are simply tunneled through the provider cloud.

5. Discovery

The capability to manually configure the addresses of the remote PEs is REQUIRED. However, the use of manual configuration is not necessary if an auto-discovery procedure is used. A number of auto-discovery procedures are compatible with this document ([RADIUS-DISC], [BGP-DISC]).

6. Control Plane

This document describes the control plane functions of signaling of PW labels. Some foundational work in the area of support for multi-homing is laid. The extensions to provide multi-homing support should work independently of the basic VPLS operation, and are not described here.

6.1. LDP Based Signaling of Demultiplexers

A full mesh of LDP sessions is used to establish the mesh of PWs. The requirement for a full mesh of PWs may result in a large number of targeted LDP sessions. Section 8 discusses the option of setting up hierarchical topologies in order to minimize the size of the VPLS full mesh.

Once an LDP session has been formed between two PEs, all PWs between these two PEs are signaled over this session.

In [PWE3-CTRL], two types of FECs are described, the PWid FEC Element (FEC type 128) and the Generalized PWid FEC Element (FEC type 129). The original FEC element used for VPLS was compatible with the PWid FEC Element. The text for signaling using PWid FEC Element has been moved to Appendix 1. What we describe below replaces that with a more generalized L2VPN descriptor, the Generalized PWid FEC Element.

6.1.1. Using the Generalized PWid FEC Element

[PWE3-CTRL] describes a generalized FEC structure that is be used for VPLS signaling in the following manner. We describe the

assignment of the Generalized PWid FEC Element fields in the context of VPLS signaling.

Control bit (C): This bit is used to signal the use of the control word as specified in [PWE3-CTRL].

PW type: The allowed PW types are Ethernet (0x0005) and Ethernet tagged mode (0x004) as specified in [IANA].

PW info length: As specified in [PWE3-CTRL].

AGI, Length, Value: The unique name of this VPLS. The AGI identifies a type of name, Length denotes the length of Value, which is the name of the VPLS. We use the term AGI interchangeably with VPLS identifier.

TAII, SAII: These are null because the mesh of PWs in a VPLS terminate on MAC learning tables, rather than on individual attachment circuits. The use of non-null TAII and SAII is reserved for future enhancements.

Interface Parameters: The relevant interface parameters are:

- MTU: the MTU (Maximum Transmission Unit) of the VPLS MUST be the same across all the PWs in the mesh.

- Optional Description String: same as [PWE3-CTRL].

- Requested VLAN ID: If the PW type is Ethernet tagged mode, this parameter may be used to signal the insertion of the appropriate VLAN ID, as defined in [PWE3-ETH].

6.2. MAC Address Withdrawal

It MAY be desirable to remove or unlearn MAC addresses that have been dynamically learned for faster convergence. This is accomplished by sending an LDP Address Withdraw Message with the list of MAC addresses to be removed to all other PEs over the corresponding LDP sessions.

We introduce an optional MAC List TLV in LDP to specify a list of MAC addresses that can be removed or unlearned using the LDP Address Withdraw Message.

The Address Withdraw message with MAC List TLVs MAY be supported in order to expedite removal of MAC addresses as the result of a topology change (e.g., failure of the primary link for a dual-homed VPLS-capable switch).

In order to minimize the impact on LDP convergence time, when the MAC list TLV contains a large number of MAC addresses, it may be

preferable to send a MAC address withdrawal message with an empty list.

6.2.1. MAC List TLV

MAC addresses to be unlearned can be signaled using an LDP Address Withdraw Message that contains a new TLV, the MAC List TLV. Its format is described below. The encoding of a MAC List TLV address is the 6-octet MAC address specified by IEEE 802 documents [g-ORIG] [802.1D-REV].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|U|F|       Type                |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC address #1                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        MAC address #1         |      MAC Address #2           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC address #2                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                               ...                             ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      MAC address #n                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        MAC address #n         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

U bit: Unknown bit. This bit MUST be set to 1. If the MAC address format is not understood, then the TLV is not understood, and MUST be ignored.

F bit: Forward bit. This bit MUST be set to 0. Since the LDP mechanism used here is targeted, the TLV MUST NOT be forwarded.

Type: Type field. This field MUST be set to 0x0404 (subject to IANA approval). This identifies the TLV type as MAC List TLV.

Length: Length field. This field specifies the total length in octets of the MAC addresses in the TLV. The length MUST be a multiple of 6.

MAC Address: The MAC address(es) being removed.

The MAC Address Withdraw Message contains a FEC TLV (to identify the VPLS affected), a MAC Address TLV and optional parameters. No optional parameters have been defined for the MAC Address Withdraw signaling. Note that if a PE receives a MAC Address Withdraw Message and does not understand it, it MUST ignore the message. In this case, instead of flushing its MAC address table, it will continue to use stale information, unless:

- it receives a packet with a known MAC address association, but from a different PW, in which case it replaces the old association, or
- it ages out the old association

The MAC Address Withdraw message only helps to speed up convergence, so PEs that do not understand the message can continue to participate in the VPLS.

6.2.2. Address Withdraw Message Containing MAC List TLV

The processing for MAC List TLV received in an Address Withdraw Message is:

For each MAC address in the TLV:
- Remove the association between the MAC address and the AC or PW over which this message is received

For a MAC Address Withdraw message with empty list:
- Remove all the MAC addresses associated with the VPLS instance (specified by the FEC TLV) except the MAC addresses learned over the PW associated with this signaling session over which the message was received

The scope of a MAC List TLV is the VPLS specified in the FEC TLV in the MAC Address Withdraw Message. The number of MAC addresses can be deduced from the length field in the TLV.

7. Data Forwarding on an Ethernet PW

This section describes the data plane behavior on an Ethernet PW used in a VPLS. While the encapsulation is similar to that described in [PWE3-ETHERNET], the functions of stripping the service-delimiting tag and using a "normalized" Ethernet frame are described.

7.1. VPLS Encapsulation actions

In a VPLS, a customer Ethernet frame without preamble is encapsulated with a header as defined in [PWE3-ETHERNET]. A customer Ethernet frame is defined as follows:

- If the frame, as it arrives at the PE, has an encapsulation that is used by the local PE as a service delimiter, i.e., to identify the customer and/or the particular service of that customer, then that encapsulation may be stripped before the frame is sent into the VPLS. As the frame exits the VPLS, the frame may have a service-delimiting encapsulation inserted.

- If the frame, as it arrives at the PE, has an encapsulation that is not service delimiting, then it is a customer frame whose encapsulation should not be modified by the VPLS. This covers, for example, a frame that carries customer-specific VLAN tags that the service provider neither knows about nor wants to modify.

As an application of these rules, a customer frame may arrive at a customer-facing port with a VLAN tag that identifies the customer's VPLS instance. That tag would be stripped before it is encapsulated in the VPLS. At egress, the frame may be tagged again, if a service-delimiting tag is used, or it may be untagged if none is used.

Likewise, if a customer frame arrives at a customer-facing port over an ATM or Frame Relay VC that identifies the customer's VPLS instance, then the ATM or FR encapsulation is removed before the frame is passed into the VPLS.

Contrariwise, if a customer frame arrives at a customer-facing port with a VLAN tag that identifies a VLAN domain in the customer L2 network, then the tag is not modified or stripped, as it belongs with the rest of the customer frame.

By following the above rules, the Ethernet frame that traverses a VPLS is always a customer Ethernet frame. Note that the two actions, at ingress and egress, of dealing with service delimiters are local actions that neither PE has to signal to the other. They allow, for example, a mix-and-match of VLAN tagged and untagged services at either end, and do not carry across a VPLS a VLAN tag that has local significance only. The service delimiter may be an MPLS label also, whereby an Ethernet PW given by [PWE3-ETHERNET] can serve as the access side connection into a PE. An RFC1483 Bridged PVC encapsulation could also serve as a service delimiter. By limiting the scope of locally significant encapsulations to the edge, hierarchical VPLS models can be developed that provide the capability to network-engineer scalable VPLS deployments, as described below.

7.2. VPLS Learning actions

Learning is done based on the customer Ethernet frame as defined above. The Forwarding Information Base (FIB) keeps track of the mapping of customer Ethernet frame addressing and the appropriate PW to use. We define two modes of learning: qualified and unqualified learning.

In unqualified learning, all the VLANs of a single customer are handled by a single VPLS, which means they all share a single broadcast domain and a single MAC address space. This means that MAC addresses need to be unique and non-overlapping among customer VLANs or else they cannot be differentiated within the VPLS

instance and this can result in loss of customer frames. An application of unqualified learning is port-based VPLS service for a given customer (e.g., customer with non-multiplexed AC where all the traffic on a physical port, which may include multiple customer VLANs, is mapped to a single VPLS instance).

In qualified learning, each customer VLAN is assigned to its own VPLS instance, which means each customer VLAN has its own broadcast domain and MAC address space. Therefore, in qualified learning, MAC addresses among customer VLANs may overlap with each other, but they will be handled correctly since each customer VLAN has its own FIB, i.e., each customer VLAN has its own MAC address space. Since VPLS broadcasts multicast frames by default, qualified learning offers the advantage of limiting the broadcast scope to a given customer VLAN. Qualified learning can result in large FIB table sizes, because the logical MAC address is now a VLAN tag + MAC address.

For STP to work in qualified learning mode, a VPLS PE must be able to forward STP BPDUs over the proper VPLS instance. In a hierarchical VPLS case (see details in Section 10), service delimiting tags (Q-in-Q or [PWE3-ETHERNET]) can be added such that PEs can unambiguously identify all customer traffic, including STP BPDUs. In a basic VPLS case, upstream switches must insert such service delimiting tags. When an access port is shared among multiple customers, a reserved VLAN per customer domain must be used to carry STP traffic. The STP frames are encapsulated with a unique provider tag per customer (as the regular customer traffic), and a PEs looks up the provider tag to send such frames across the proper VPLS instance.

8. Data Forwarding on an Ethernet VLAN PW

This section describes the data plane behavior on an Ethernet VLAN PW in a VPLS. While the encapsulation is similar to that described in [PWE3-ETHERNET], the functions of imposing tags and using a "normalized" Ethernet frame are described. The learning behavior is the same as for Ethernet PWs.

8.1. VPLS Encapsulation actions

In a VPLS, a customer Ethernet frame without preamble is encapsulated with a header as defined in [PWE3-ETHERNET]. A customer Ethernet frame is defined as follows:

- If the frame, as it arrives at the PE, has an encapsulation that is part of the customer frame, and is also used by the local PE as a service delimiter, i.e., to identify the customer and/or the particular service of that customer, then that encapsulation is preserved as the frame is sent into the VPLS, unless the Requested VLAN ID optional parameter was

signaled. In that case, the VLAN tag is overwritten before the frame is sent out on the PW.

- If the frame, as it arrives at the PE, has an encapsulation that does not have the required VLAN tag, a null tag is imposed if the Requested VLAN ID optional parameter was not signaled.

As an application of these rules, a customer frame may arrive at a customer-facing port with a VLAN tag that identifies the customer's VPLS instance and also identifies a customer VLAN. That tag would be preserved as it is encapsulated in the VPLS.

The Ethernet VLAN PW provides a simple way to preserve customer 802.1p bits.

A VPLS MAY have both Ethernet and Ethernet VLAN PWs. However, if a PE is not able to support both PWs simultaneously, it SHOULD send a Label Release on the PW messages that it cannot support with a status code "Unknown FEC" as given in [RFC3036].

9. Operation of a VPLS

We show here, in Figure 2 below, an example of how a VPLS works. The following discussion uses the figure below, where a VPLS has been set up between PE1, PE2 and PE3. The VPLS connects a customer with 4 sites labeled A1, A2, A3 and A4 through CE1, CE2, CE3 and CE4, respectively.

Initially, the VPLS is set up so that PE1, PE2 and PE3 have a full mesh of Ethernet PWs. The VPLS instance is assigned a identifier (AGI). For the above example, say PE1 signals PW label 102 to PE2 and 103 to PE3, and PE2 signals PW label 201 to PE1 and 203 to PE3.

```
                                                         -----
                                                        /  A1 \
           ----                                    ----CE1    |
          /    \          --------       -------  /     |     |
          | A2 CE2-      /        \     /       PE1     \     /
          \    /   \    /          \---/         \       -----
           ----     ---PE2                        |
                       | Service Provider Network |
                        \          /   \         /
                 -----  PE3       /     \       /
                 |Agg|_/  --------       -------
                -|   |
         ----  / -----  ----
        /    \/    \   /    \             CE = Customer Edge Router
        | A3 CE3    -CE4 A4 |             PE = Provider Edge Router
        \    /         \    /             Agg = Layer 2 Aggregation
         ----           ----
```

Figure 2: Example of a VPLS

Assume a packet from A1 is bound for A2. When it leaves CE1, say it has a source MAC address of M1 and a destination MAC of M2. If PE1 does not know where M2 is, it will flood the packet, i.e., send it to PE2 and PE3. When PE2 receives the packet, it will have a PW label of 201. PE2 can conclude that the source MAC address M1 is behind PE1, since it distributed the label 201 to PE1. It can therefore associate MAC address M1 with PW label 102.

9.1. MAC Address Aging

PEs that learn remote MAC addresses SHOULD have an aging mechanism to remove unused entries associated with a PW label. This is important both for conservation of memory as well as for administrative purposes. For example, if a customer site A is shut down, eventually, the other PEs should unlearn A's MAC address.

The aging timer for MAC address M SHOULD be reset when a packet with source MAC address M is received.

10. A Hierarchical VPLS Model

The solution described above requires a full mesh of tunnel LSPs between all the PE routers that participate in the VPLS service. For each VPLS service, n*(n-1)/2 PWs must be setup between the PE routers. While this creates signaling overhead, the real detriment to large scale deployment is the packet replication requirements for each provisioned PWs on a PE router. Hierarchical connectivity, described in this document reduces signaling and replication overhead to allow large scale deployment.

In many cases, service providers place smaller edge devices in multi-tenant buildings and aggregate them into a PE in a large Central Office (CO) facility. In some instances, standard IEEE 802.1q (Dot 1Q) tagging techniques may be used to facilitate mapping CE interfaces to VPLS access circuits at a PE.

It is often beneficial to extend the VPLS service tunneling techniques into the access switch domain. This can be accomplished by treating the access device as a PE and provisioning PWs between it and every other edge, as a basic VPLS. An alternative is to utilize [PWE3-ETHERNET] PWs or Q-in-Q logical interfaces between the access device and selected VPLS enabled PE routers. Q-in-Q encapsulation is another form of L2 tunneling technique, which can be used in conjunction with MPLS signaling as will be described later. The following two sections focus on this alternative approach. The VPLS core PWs (hub) are augmented with access PWs (spoke) to form a two-tier hierarchical VPLS (H-VPLS).

Spoke PWs may be implemented using any L2 tunneling mechanism, expanding the scope of the first tier to include non-bridging VPLS PE routers. The non-bridging PE router would extend a spoke PW

from a Layer-2 switch that connects to it, through the service core network, to a bridging VPLS PE router supporting hub PWs. We also describe how VPLS-challenged nodes and low-end CEs without MPLS capabilities may participate in a hierarchical VPLS.

For rest of this discussion we refer to a bridging capable access device as MTU-s and a non-bridging capable PE as PE-r. We refer to a routing and bridging capable device as PE-rs.

10.1. Hierarchical connectivity

This section describes the hub and spoke connectivity model and describes the requirements of the bridging capable and non-bridging MTU-s devices for supporting the spoke connections.

10.1.1. Spoke connectivity for bridging-capable devices

In Figure 3 below, three customer sites are connected to an MTU-s through CE-1, CE-2, and CE-3. The MTU-s has a single connection (PW-1) to PE1-rs. The PE-rs devices are connected in a basic VPLS full mesh. For each VPLS service, a single spoke PW is set up between the MTU-s and the PE-rs based on [PWE3-CTRL]. Unlike traditional PWs that terminate on a physical (or a VLAN-tagged logical) port, a spoke PW terminates on a virtual switch instance (VSI, see [L2FRAME]) on the MTU-s and the PE-rs devices.

```
                                                          PE2-rs
                                                        +--------+
                                                        |        |
                                                        |   --   |
                                                        |  /  \  |
    CE-1                                                |  \S /  |
     \                                                  |   --   |
      \                                                 +--------+
       \   MTU-s                          PE1-rs        /   |
        +--------+                      +--------+     /    |
        |        |                      |        |    /     |
        |   --   |      PW-1            |   --   |---/      |
        |  /  \--|- - - - - - - - - - - |  /  \  |          |
        |  \S /  |                      |  \S /  |          |
        |   --   |                      |   --   |---\      |
        +--------+                      +--------+    \     |
         /                                             \    |
       ----                                             +--------+
      |Agg |                                            |        |
       ----                                             |  --    |
      /    \                                            | /  \   |
     CE-2  CE-3                                         | \S /   |
                                                        |  --    |
                                                        +--------+
                                                          PE3-rs

    Agg = Layer-2 Aggregation
    --
   /  \
   \S / = Virtual Switch Instance
    --
```

Figure 3: An example of a hierarchical VPLS model

The MTU-s and the PE-rs treat each spoke connection like an AC of the VPLS service. The PW label is used to associate the traffic from the spoke to a VPLS instance.

10.1.1.1. MTU-s Operation

An MTU-s is defined as a device that supports layer-2 switching functionality and does all the normal bridging functions of learning and replication on all its ports, including the spoke, which is treated as a virtual port. Packets to unknown destinations are replicated to all ports in the service including the spoke. Once the MAC address is learned, traffic between CE1 and CE2 will be switched locally by the MTU-s saving the capacity of the spoke to the PE-rs. Similarly traffic between CE1 or CE2 and any remote destination is switched directly on to the spoke and sent to the PE-rs over the point-to-point PW.

Since the MTU-s is bridging capable, only a single PW is required per VPLS instance for any number of access connections in the same

VPLS service. This further reduces the signaling overhead between the MTU-s and PE-rs.

If the MTU-s is directly connected to the PE-rs, other encapsulation techniques such as Q-in-Q can be used for the spoke.

10.1.1.2. PE-rs Operation

A PE-rs is a device that supports all the bridging functions for VPLS service and supports the routing and MPLS encapsulation, i.e., it supports all the functions described for a basic VPLS as described above.

The operation of PE-rs is independent of the type of device at the other end of the spoke. Thus, the spoke from the MTU-s is treated as a virtual port and the PE-rs will switch traffic between the spoke PW, hub PWs, and ACs once it has learned the MAC addresses.

10.1.2. Advantages of spoke connectivity

Spoke connectivity offers several scaling and operational advantages for creating large scale VPLS implementations, while retaining the ability to offer all the functionality of the VPLS service.

- Eliminates the need for a full mesh of tunnels and full mesh of PWs per service between all devices participating in the VPLS service.
- Minimizes signaling overhead since fewer PWs are required for the VPLS service.
- Segments VPLS nodal discovery. MTU-s needs to be aware of only the PE-rs node although it is participating in the VPLS service that spans multiple devices. On the other hand, every VPLS PE-rs must be aware of every other VPLS PE-rs and all of its locally connected MTU-s and PE-r devices.
- Addition of other sites requires configuration of the new MTU-s but does not require any provisioning of the existing MTU-s devices on that service.
- Hierarchical connections can be used to create VPLS service that spans multiple service provider domains. This is explained in a later section.

Note that as more devices participate in the VPLS, there are more devices that require the capability for learning and replication.

10.1.3. Spoke connectivity for non-bridging devices

In some cases, a bridging PE-rs may not be deployed, or a PE-r might already have been deployed. In this section, we explain how a PE-r that does not support any of the VPLS bridging functionality can participate in the VPLS service.

In Figure 4, three customer sites are connected through CE-1, CE-2 and CE-3 to the VPLS through PE-r. For every attachment circuit that participates in the VPLS service, PE-r creates a point-to-point PW that terminates on the VSI of PE1-rs.

```
                                                          PE2-rs
                                                        +--------+
                                                        |        |
                                                        |   --   |
                                                        |  /  \  |
    CE-1                                                |  \S /  |
     \                                                  |   --   |
      \                                                 +--------+
       \   PE-r                          PE1-rs         /   |
        +--------+                      +--------+     /    |
        |\       |                      |        |    /     |
        | \      |      PW-1            |   --   |---/      |
        |  ------|- - - - - - - - - - - |  /  \  |          |
        |   -----|- - - - - - - - - - - |  \S /  |          |
        |  /     |                      |   --   |---\      |
        +--------+                      +--------+    \     |
         /                                             \    |
       ----                                            +--------+
      | Agg|                                           |        |
       ----                                            |   --   |
      /    \                                           | /  \   |
     CE-2  CE-3                                        | \S /   |
                                                       |   --   |
                                                       +--------+
                                                         PE3-rs
```

Figure 4: An example of a hierarchical VPLS
with non-bridging spokes

The PE-r is defined as a device that supports routing but does not support any bridging functions. However, it is capable of setting up PWs between itself and the PE-rs. For every port that is supported in the VPLS service, a PW is setup from the PE-r to the PE-rs. Once the PWs are setup, there is no learning or replication function required on the part of the PE-r. All traffic received on any of the ACs is transmitted on the PW. Similarly all traffic received on a PW is transmitted to the AC where the PW terminates. Thus traffic from CE1 destined for CE2 is switched at PE1-rs and not at PE-r.

Note that in the case where PE-r devices use Provider VLANs (P-VLAN) as demultiplexers instead of PWs, PE1-rs can treat them as such and map these "circuits" into a VPLS domain to provide bridging support between them.

This approach adds more overhead than the bridging capable (MTU-s) spoke approach since a PW is required for every AC that

participates in the service versus a single PW required per service (regardless of ACs) when an MTU-s is used. However, this approach offers the advantage of offering a VPLS service in conjunction with a routed internet service without requiring the addition of new MTU-s.

10.2. Redundant Spoke Connections

An obvious weakness of the hub and spoke approach described thus far is that the MTU-s has a single connection to the PE-rs. In case of failure of the connection or the PE-rs, the MTU-s suffers total loss of connectivity.

In this section we describe how the redundant connections can be provided to avoid total loss of connectivity from the MTU-s. The mechanism described is identical for both, MTU-s and PE-r devices.

10.2.1. Dual-homed MTU-s

To protect from connection failure of the PW or the failure of the PE-rs, the MTU-s or the PE-r is dual-homed into two PE-rs devices. The PE-rs devices must be part of the same VPLS service instance.

```
                                                          PE2-rs
                                                        +--------+
                                                        |        |
                                                        |   --   |
                                                        |  /  \  |
    CE-1                                                |  \S /  |
      \                                                 |   --   |
       \                                                +--------+
        \  MTU-s                          PE1-rs        /   |
        +--------+                      +--------+     /    |
        |        |                      |        |    /     |
        |   --   |   Primary PW         |   --   |---/      |
        |  /  \  |- - - - - - - - - - - |  /  \  |          |
        |  \S /  |                      |  \S /  |          |
        |   --   |                      |   --   |---\      |
        +--------+                      +--------+    \     |
          /      \                                     \    |
         /        \                                     +--------+
        /          \                                    |        |
       CE-2         \                                   |  --    |
                     \     Secondary PW                 | /  \   |
                      - - - - - - - - - - - - - - - - - | \S /   |
                                                        |  --    |
                                                        +--------+
                                                          PE3-rs
```

Figure 5: An example of a dual-homed MTU-s

In Figure 5, two customer sites are connected through CE-1 and CE-2 to an MTU-s. The MTU-s sets up two PWs (one each to PE1-rs and PE3-rs) for each VPLS instance. One of the two PWs is designated as

primary and is the one that is actively used under normal conditions, while the second PW is designated as secondary and is held in a standby state. The MTU-s negotiates the PW labels for both the primary and secondary PWs, but does not use the secondary PW unless the primary PW fails. How a spoke is designated primary or secondary is outside of the scope of this document. For example, a spanning tree instance running between only the MTU-s and the two PE-rs nodes is one possible method. Another method could be configuration.

10.2.2. Failure detection and recovery

The MTU-s should control the usage of the spokes to the PE-rs devices. If the spokes are PWs, then LDP signaling is used to negotiate the PW labels, and the hello messages used for the LDP session could be used to detect failure of the primary PW. The use of other mechanisms which could provide faster detection failures is outside the scope of this document.

Upon failure of the primary PW, MTU-s immediately switches to the secondary PW. At this point the PE3-rs that terminates the secondary PW starts learning MAC addresses on the spoke PW. All other PE-rs nodes in the network think that CE-1 and CE-2 are behind PE1-rs and may continue to send traffic to PE1-rs until they learn that the devices are now behind PE3-rs. The unlearning process can take a long time and may adversely affect the connectivity of higher level protocols from CE1 and CE2. To enable faster convergence, the PE3-rs where the secondary PW got activated may send out a flush message (as explained in section 4.2), using the MAC List TLV as defined in Section 6, to all PE-rs nodes. Upon receiving the message, PE-rs nodes flush the MAC addresses associated with that VPLS instance.

10.3. Multi-domain VPLS service

Hierarchy can also be used to create a large scale VPLS service within a single domain or a service that spans multiple domains without requiring full mesh connectivity between all VPLS capable devices. Two fully meshed VPLS networks are connected together using a single LSP tunnel between the VPLS "border" devices. A single spoke PW per VPLS service is set up to connect the two domains together.

When more than two domains need to be connected, a full mesh of inter-domain spokes is created between border PEs. Forwarding rules over this mesh are identical to the rules defined in section 5.

This creates a three-tier hierarchical model that consists of a hub-and-spoke topology between MTU-s and PE-rs devices, a full-mesh topology between PE-rs, and a full mesh of inter-domain spokes between border PE-rs devices.

This document does not specify how redundant border PEs per domain per VPLS instance can be supported.

11. Hierarchical VPLS model using Ethernet Access Network

In this section the hierarchical model is expanded to include an Ethernet access network. This model retains the hierarchical architecture discussed previously in that it leverages the full-mesh topology among PE-rs devices; however, no restriction is imposed on the topology of the Ethernet access network (e.g., the topology between MTU-s and PE-rs devices is not restricted to hub and spoke).

The motivation for an Ethernet access network is that Ethernet-based networks are currently deployed by some service providers to offer VPLS services to their customers. Therefore, it is important to provide a mechanism that allows these networks to integrate with an IP or MPLS core to provide scalable VPLS services.

One approach of tunneling a customer's Ethernet traffic via an Ethernet access network is to add an additional VLAN tag to the customer's data (which may be either tagged or untagged). The additional tag is referred to as Provider's VLAN (P-VLAN). Inside the provider's network each P-VLAN designates a customer or more specifically a VPLS instance for that customer. Therefore, there is a one-to-one correspondence between a P-VLAN and a VPLS instance. In this model, the MTU-s needs to have the capability of adding the additional P-VLAN tag to non-multiplexed ACs where customer VLANs are not used as service delimiters. This functionality is described in [802.1ad].

If customer VLANs need to be treated as service delimiters (e.g., the AC is a multiplexed port), then the MTU-s needs to have the additional capability of translating a customer VLAN (C-VLAN) to a P-VLAN, or push an additional P-VLAN tag, in order to resolve overlapping VLAN tags used by different customers. Therefore, the MTU-s in this model can be considered as a typical bridge with this additional capability. This functionality is described in [802.1ad].

The PE-rs needs to be able to perform bridging functionality over the standard Ethernet ports toward the access network as well as over the PWs toward the network core. In this model, the PE-rs may need to run STP towards the access network, in addition to split-horizon over the MPLS core. The PE-rs needs to map a P-VLAN to a VPLS-instance and its associated PWs and vice versa.

The details regarding bridge operation for MTU-s and PE-rs (e.g., encapsulation format for Q-in-Q messages, customer's Ethernet control protocol handling, etc.) are outside of the scope of this document and they are covered in [802.1ad]. However, the relevant part is the interaction between the bridge module and the MPLS/IP PWs in the PE-rs, which behaves just as in a regular VPLS.

11.1. Scalability

Since each P-VLAN corresponds to a VPLS instance, the total number of VPLS instances supported is limited to 4K. The P-VLAN serves as a local service delimiter within the provider's network that is stripped as it gets mapped to a PW in a VPLS instance. Therefore, the 4K limit applies only within an Ethernet access network (Ethernet island) and not to the entire network. The SP network consists of a core MPLS/IP network that connects many Ethernet islands. Therefore, the number of VPLS instances can scale accordingly with the number of Ethernet islands (a metro region can be represented by one or more islands).

11.2. Dual Homing and Failure Recovery

In this model, an MTU-s can be dual homed to different devices (aggregators and/or PE-rs devices). The failure protection for access network nodes and links can be provided through running STP in each island. The STP of each island is independent from other islands and do not interact with each other. If an island has more than one PE-rs, then a dedicated full-mesh of PWs is used among these PE-rs devices for carrying the SP BPDU packets for that island. On a per P-VLAN basis, STP will designate a single PE-rs to be used for carrying the traffic across the core. The loop-free protection through the core is performed using split-horizon and the failure protection in the core is performed through standard IP/MPLS re-routing.

12. Contributors

Loa Andersson, TLA
Ron Haberman, Alcatel
Juha Heinanen, Independent
Giles Heron, Tellabs
Sunil Khandekar, Alcatel
Luca Martini, Cisco
Pascal Menezes, Independent
Rob Nath, Riverstone
Eric Puetz, SBC
Vasile Radoaca, Nortel
Ali Sajassi, Cisco
Yetik Serbest, SBC
Nick Slabakov, Riverstone
Andrew Smith, Consultant
Tom Soon, SBC
Nick Tingle, Alcatel

13. Acknowledgments

We wish to thank Joe Regan, Kireeti Kompella, Anoop Ghanwani, Joel Halpern, Rick Wilder, Jim Guichard, Steve Phillips, Norm Finn, Matt

Squire, Muneyoshi Suzuki, Waldemar Augustyn, Eric Rosen, Yakov Rekhter, Sasha Vainshtein, and Du Wenhua for their valuable feedback.

We would also like to thank Rajiv Papneja (ISOCORE), Winston Liu (Ixia), and Charlie Hundall for identifying issues with the draft in the course of the interoperability tests.

We would also like to thank Ina Minei, Bob Thomas, Eric Gray and Dimitri Papadimitriou for their thorough technical review of the document.

14. Security Considerations

A more comprehensive description of the security issues involved in L2VPNs is covered in [VPN-SEC]. An unguarded VPLS service is vulnerable to some security issues which pose risks to the customer and provider networks. Most of the security issues can be avoided through implementation of appropriate guards. A couple of them can be prevented through existing protocols.

- Data plane aspects
    - Traffic isolation between VPLS domains is guaranteed by the use of per VPLS L2 FIB table and the use of per VPLS PWs
    - The customer traffic, which consists of Ethernet frames, is carried unchanged over VPLS. If security is required, the customer traffic SHOULD be encrypted and/or authenticated before entering the service provider network
    - Preventing broadcast storms can be achieved by using routers as CPE devices or by rate policing the amount of broadcast traffic that customers can send
- Control plane aspects
    - LDP security (authentication) methods as described in [RFC-3036] SHOULD be applied. This would prevent unauthenticated messages from disrupting a PE in a VPLS
- Denial of service attacks
    - Some means to limit the number of MAC addresses (per site per VPLS) that a PE can learn SHOULD be implemented

15. IANA Considerations

The type field in the MAC List TLV is defined as 0x404 in section 6.2.1 and is subject to IANA approval.

16. References

16.1. Normative References

[PWE3-ETHERNET] "Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks", draft-ietf-pwe3-ethernet-encap-10.txt, Work in progress, June 2005.

[PWE3-CTRL] "Transport of Layer 2 Frames over MPLS", draft-ietf-pwe3-control-protocol-17.txt, Work in progress, June 2005.

[802.1D-ORIG] Original 802.1D - ISO/IEC 10038, ANSI/IEEE Std 802.1D-1993 "MAC Bridges".

[802.1D-REV] 802.1D - "Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Access Control (MAC) Bridges: Revision. This is a revision of ISO/IEC 10038: 1993, 802.1j-1992 and 802.6k-1992. It incorporates P802.11c, P802.1p and P802.12e." ISO/IEC 15802-3: 1998.

[802.1Q] 802.1Q - ANSI/IEEE Draft Standard P802.1Q/D11, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks", July 1998.

[RFC3036] "LDP Specification", L. Andersson, et al., RFC 3036, January 2001.

[IANA] "IANA Allocations for pseudo Wire Edge to Edge Emulation (PWE3)" Martini,Townsley, draft-ietf-pwe3-iana-allocation-08.txt, Work in progress, February 2005.

16.2. Informative References

[BGP-VPN] "BGP/MPLS VPNs", draft-ietf-l3vpn-rfc2547bis-03.txt, Work in Progress, October 2004.

[RADIUS-DISC] "Using Radius for PE-Based VPN Discovery", draft-ietf-l2vpn-radius-pe-discovery-01.txt, Work in Progress, February 2005.

[BGP-DISC] "Using BGP as an Auto-Discovery Mechanism for Network-based VPNs", draft-ietf-l3vpn-bgpvpn-auto-06.txt, Work in Progress, June 2005.

[L2FRAME] "Framework for Layer 2 Virtual Private Networks (L2VPNs)", draft-ietf-l2vpn-l2-framework-05, Work in Progress, June 2004.

[L2VPN-REQ] "Service Requirements for Layer-2 Provider Provisioned Virtual Private Networks", draft-ietf-l2vpn-requirements-04.txt, Work in Progress, October 2005.

[VPN-SEC] "Security Framework for Provider Provisioned Virtual Private Networks", draft-ietf-l3vpn-security-framework-03.txt, Work in Progress, November 2004.

[802.1ad] "IEEE standard for Provider Bridges", Work in Progress, December 2002.

17. Appendix: VPLS Signaling using the PWid FEC Element

This section is being retained because live deployments use this version of the signaling for VPLS.

The VPLS signaling information is carried in a Label Mapping message sent in downstream unsolicited mode, which contains the following PWid FEC TLV.

PW, C, PW Info Length, Group ID, Interface parameters are as defined in [PWE3-CTRL].

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    PW TLV     |C|         PW Type             |PW info Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Group ID                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        PWID                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Interface parameters                    |
~                                                               ~
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

We use the Ethernet PW type to identify PWs that carry Ethernet traffic for multipoint connectivity.

In a VPLS, we use a VCID (which, when using the PWid FEC, has been substituted with a more general identifier (AGI), to address extending the scope of a VPLS) to identify an emulated LAN segment. Note that the VCID as specified in [PWE3-CTRL] is a service identifier, identifying a service emulating a point-to-point virtual circuit. In a VPLS, the VCID is a single service identifier, so it has global significance across all PEs involved in the VPLS instance.

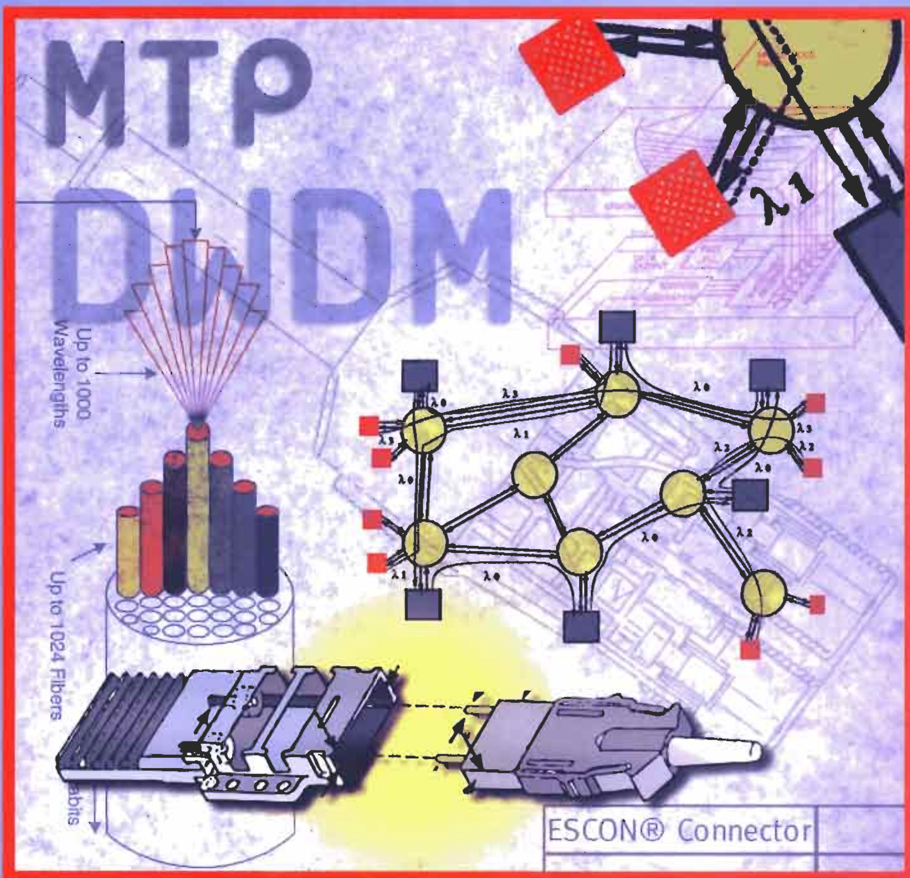
18. Authors' Addresses

Marc Lasserre
Riverstone Networks
Email: marc@riverstonenet.com

Vach Kompella
Alcatel
Email: vach.kompella@alcatel.com

EXHIBIT 4C

Academic Press



Fiber Optic DATA COMMUNICATION

Technological Trends and Advances

Edited by Casimer DeCusatis

FIBER OPTIC DATA COMMUNICATION: TECHNOLOGICAL TRENDS AND ADVANCES

FIBER OPTIC DATA COMMUNICATION: TECHNOLOGICAL TRENDS AND ADVANCES

CASIMER DeCUSATIS
Editor
IBM Corporation
Poughkeepsie, New York

ACADEMIC PRESS

An Elsevier Science Imprint

San Diego London Boston
New York Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

ACADEMIC PRESS
An Elsevier Science Imprint
525 B Street, Suite 1900, San Diego, CA 92101-4495, USA
http://academicpress.com

ACADEMIC PRESS LIMITED
An Elsevier Science Imprint
Harcourt Place, 32 Jamestown Road, London NW1 7BY, UK
http://academicpress.com

Library of Congress Catalog Card Number: 2001095439
International Standard Book Number: 0-12-207892-6

Printed in China
02 03 04 05 RDC 9 8 7 6 5 4 3 2 1

To the people who give meaning to my life
and taught me to look for wonder in the world:
my wife, Carolyn, my daughters, Anne and Rebecca, my parents,
my godmother, Isabel, and her mother, Mrs. Crease. — CD

Contents

Contributors

Numbers in parentheses indicate the pages on which the authors' contributions begin.

Prashant Chouta (303), Cookson Performance Solutions, 25 Forbes Boulevard, Foxborough, Massachusetts 02053

Casimer DeCusatis (63, 89, 134, 270), IBM Corporation, 2455 South Road MS P343, Poughkeepsie, New York 12601

John Fox (63, 89), ComputerCrafts, Inc., 57 Thomas Road, Hawthorne, New Jersey 07507

Ali Ghiasi (321), Broadcom Corporation (formerly SUN Microsystems), 19947 Linden Brook Lane, Cupertino, California 95014

Ronald C. Lasky (32, 303), Consultant, 26 Howe Street, Medway, Massachusetts 02053

Chung-Sheng Li (422), IBM Thomas J. Watson Research Center, 30 Sawmill River Road, Hawthorne, New York 10532

Daniel M. Litynski (352), College of Engineering and Applied Sciences, Western Michigan University, 2022 Kohrman Hall, Kalamazoo, Michigan 49008

Eric Maass (447), Motorola, Incorporated, 2100 Elliot Road, Tempe, Arizona 85284

Rainer Michalzik (216), University of Ulm, Optoelectronics Dept., Albert-Einstein-Allee 45, D-89069 Ulm, Germany

Jeff D. Montgomery (3), ElectroniCast Corporation, 800 South Claremont St., San Mateo, California 94402

Yann Y. Morvan (32), Cookson Electronics, New Haven, Connecticut, 06510

Andre H. Sayles (352), Photonics Research Center and Department of Electrical Engineering and Computer Science, U.S. Military Academy, West Point, New York 10996

David B. Sher (270), Mathematics/Statistics/CMP Dept., Nassau Community College, 1 Education Drive, Garden City, New York 11530

Barry L. Shoop (352), Photonics Research Center and Department of Electrical Engineering and Computer Science, U.S. Military Academy, West Point, New York 10996

Adam Singer (303), Cookson Performance Solutions, 25 Forbes Boulevard, Foxborough, Massachusetts 02053

Preface

"I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year."

—Attributed to the chief editor for business books, Prentice Hall, 1957

"There is nothing more difficult to take in hand, more perilous to conduct, or more uncertain in its success, than to take the lead in the introduction of a new order of things."

—Machiavelli

This book arose during the process of revising the second edition of the Handbook of Fiber Optic Data Communication, when it became apparent that one book wasn't enough to contain all of the technology developments in wavelength multiplexing, optically clustered servers, small form factor transceivers and connectors, and other emerging technologies. As a result, we decided to split off the four chapters on futures from the original handbook, combine them with many new chapters, and form this book, which can serve as either a companion to the original book or a stand-alone reference volume.

Many new chapters have also been added to address the rapidly accelerating rate of change that has characterized this field. New component technologies for optical backplanes, parallel coupled computer architectures, and smart pixels are among the topics covered here. Open standards, which to a great extent have created the Internet and the Web (remember TCP/IP?) also continue to evolve, and new standards are emerging to deal with the requirements of the next generation intelligent optical infrastructure; some of these standards, such as Infiniband, are covered in this volume. There are also new chapters on the history of communications technology (with apologies to those who have noted that it remains

difficult to determine exactly who invented the first one of anything, and that the history of science is filled with tales of misplaced credit), and predictions of the future, as envisioned by some of the leading commercial technology forecasters. Given the rapid and accelerating rate of change in this field, it is inevitable that some topics of interest will not be covered. As this book goes to press, for example, there is growing interest in microelectromechanical devices (MEMs) and so-called micro-photonics, ultra high data rate transceivers (40 Gbit/s and above), advanced storage area networks and network attached storage, and other areas that are beyond the scope of this text. It is our hope that these and other topics will be incorporated into future editions of this book, just as the original Handbook of Fiber Optic Data Communication has grown through the years.

An undertaking such as this would not be possible without the concerted efforts of many contributing authors and a supportive staff at the publisher, to all of whom I extend my deepest gratitude. The following associate editors contributed to the first edition of the Handbook of Fiber Optic Data Communication: Eric Maass, Darrin Clement, and Ronald Lasky. As always, this book is dedicated to my parents, who first helped me see the wonder in the world; to the memory of my godmother Isabel; and to my wife, Carolyn, and daughters Anne and Rebecca, without whom this work would not have been possible.

Dr. Casimer DeCusatis, Editor
Poughkeepsie, New York

Part 1 Technology Advances

Chapter 1 | History of Fiber Optics

Jeff D. Montgomery

Chairman/Founder, ElectroniCast Corporation, San Mateo, California 94402

In this review of the history of communication via fiber optics, we examine this relatively recent advancement within the context of communication through history. We also offer projections of where this continuing advancement in communication technology may lead us over the next half century.

1.1. Earliest Civilization to the Printing Press

1.1.1. COMMUNICATION THROUGH THE AGES

All species communicate within their group. The evolution of the human species, however, appears to have been much more rapid and dramatic than the evolution of other species. This human advancement has coincided with an increasingly rapid advancement in communication capability. Is this merely a coincidence, or is there a causal relationship?

The earliest human communication, we assume, was vocal; a capability shared by numerous other species. Archaeological information, however, indicates that, tens of thousands of years ago, humans also began to communicate via stored information in addition to the vocal mode. Cave paintings and cliffside carvings have survived over time, to now, conveying information that at the time was useful. Findings also indicate signal fires existed in those early times, to transmit (via light) information, presumably the

FIBER OPTIC DATA COMMUNICATION:
TECHNOLOGICAL TRENDS AND ADVANCES
$35.00

ISBN: 0-12-207892-6

sighting of the approach of other humans, the appearance of game animals, or other intelligence. Smoke signal communication emerged along with nautical signal flags, followed by light-beam and flag semaphores.

As civilization advanced (and humans apparently became much more numerous), communication became increasingly complex. Symbols to represent items of interest were conceived and adopted. Techniques were developed to carve these symbols in stone, or to paint them onto media such as walls or sheets made from papyrus reeds — the early communication storage media. The papyrus-enscribed messages were especially significant, in that they were transportable — the early telecommunication ("communication at a distance").

While the development of symbols and media was a major advancement, there were still some major handicaps. Carved messages, in particular, had very low portability. A more general problem was that forming the symbols into the media was a high-level skill that required years of training. Kings and common people could not write (and, in general, could not read). Beyond the limited number of scribes available, and the relatively high cost per message inscribed, was the time required to complete a message; hours to days for a simple scroll; lifetimes for stone carvings. Also, each copy, if wanted, required as much effort and time as the original. These general techniques, however, did not change dramatically over a span of thousands of years. A degree of "shorthand" symbols were developed for commercial messages, and the language became richer through development of more and increasingly refined symbols. Still, it remained a slow form of communication, limited to royalty, wealthy merchants, military leaders, and scholars.

As the need for copies of messages, such as distribution of proclamations, increased, entrepreneurs developed the technique of transferring a symbolic message from the original by applying ink and transferring the message to another surface. Printing! Naturally, as this technique evolved, message originators also evolved to sending out more copies. There also naturally evolved a tendency to create longer, more complex messages. So, although making multiple copies became feasible, crafting the original print master remained the role of a master craftsman and, as messages became longer, more time was required.

Within this period, some messages became long enough to be "books." Creating the print master for a book occupied a crew of engravers for many years. Although communication certainly was advancing, it remained expensive and slow to initiate in transportable, storable form.

1.2. The Next 500 Years: Printing Press to Year 2000

1.2.1. PRINTING PRESS CHANGES THE RULES

The invention of the movable-type printing press by J. Gutenberg, circa 1450, was a major breakthrough. By this time, the language of communication had evolved from pictorial symbols to words formed from a set of characters or other symbols. These were laboriously engraved into printing plates, requiring days to years per plate. With the availability of movable type pieces that could be arranged to construct a clamped-together plate, the time to create a plate was reduced by orders of magnitude; from days to minutes. Of equal importance, the plates could now be constructed by a technician having relatively modest training, instead of by a skilled artisan with years of training and apprenticeship. With the Gutenberg press, the cost of books could be greatly reduced, becoming financially available to a much larger segment of the populace. Over the ensuing 400 years, instruction books became widely available to all students, current news publication flourished, and entertainment books emerged.

1.2.2. THE CASCADE OF INVENTION

With the evolution of the printing press, the worldwide exchange of information between scholars, inventors, and other innovators accelerated. Especially over the most recent two centuries, significant inventions cascaded, often standing on the shoulders of earlier inventions. Some of the key inventions related to the advancement of communication are noted in Fig. 1.1. Signal transmission through space by electromagnetics (Marconi), electrical conductance principles (Maxwell), mechanized digital computing (Babbage), the telephone (Bell) were landmark inventions that set the platforms for the just-completed Magnetic Century. Vacuum tube amplifiers and rectifiers emerged, making radio transmission and reception feasible (and, ultimately, ubiquitous and affordable). Electronic computing evolved, mid-century, from an interesting intellectual concept to become a tool, albeit very expensive, for controlling massive electrical power grids and for tackling otherwise overwhelmingly challenging scientific calculations. (It was visualized that several of these machines, perhaps dozens, might ultimately be useful worldwide; Thomas J. Watson, International Business Machines Chairman, postulated a potential worldwide market for perhaps five of their computing machines.)

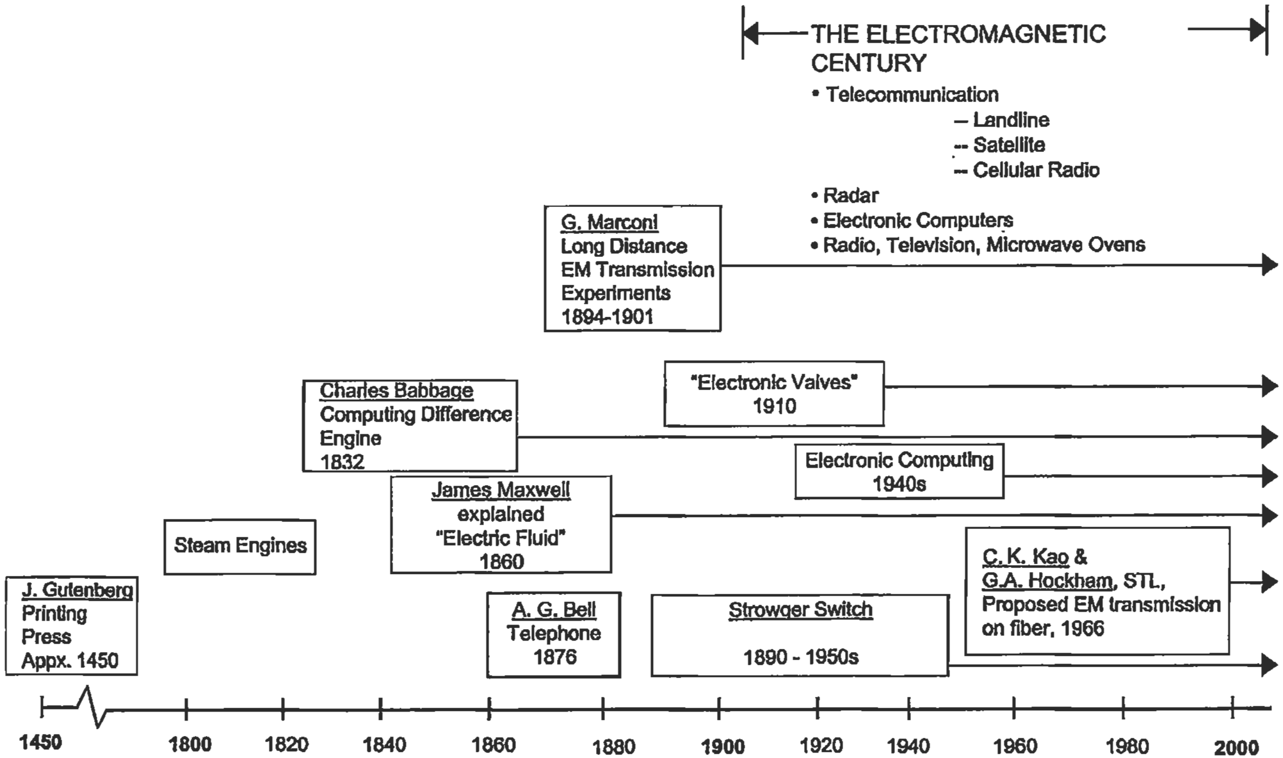


Fig. 1.1 The electromagnetic century.

Over the 1633–1882 span, mechanical computation machines were of continuing interest, with concepts developed by Pascal, Leibnitz, and Schickhard, culminating in the first serious effort to build a mechanical calculator machine (by Charles Babbage, in 1882). The first working electromechanical calculator was built by IBM engineers in 1930 (the IBM Automatic Sequence Controlled Calculator, Mark I), under the direction of Professor Aiken of Harvard University. [1] The first electronic calculator, ENIAC, was built by Eckert and Mauchly, of the University of Pennsylvania, in 1946.

The Strowger switch, invented within Bell Laboratories, illustrates a significant point that keeps recurring in the evolution of communication (and in other fields): When a problem evolves and advances to the point that it threatens the continuing evolution of an important field, inventive minds find a feasible solution. The early wire-line telephone systems required switching, to connect a specific originating telephone to the desired other telephone instrument. This was done by an operator who received verbal instructions from the originator, then plugged a connection cord between the two appropriate receptacles on the switchboard. As the number of subscribers and the number of calls per subscriber steadily increased, it became apparent that within a relatively few years it would no longer be feasible to recruit enough operators to do the switching. Thus, the Strowger switch, doing the same task based on telephone-number-based electrical signals, was developed. This switch occupied a lot less physical space, and did the task faster, at less cost, and with higher 24-hour-per-day dependability and accuracy. The Strowger switch, introduced in the late 1800s, bridged the transition into the Electromagnetic Century.

The advancement of telecommunication technology and facilities was especially dramatic through the first half of the 20th century. Telephone communication advanced from two-wire lines to hundreds of parallel voice grade lines, as illustrated in Fig. 1.2, colliding with another roadblock. The number of open, uninsulated lines routed along city streets and into major office buildings approached the physical space limits. This drove network developers to evolve to "twisted pair" insulated copper wires that greatly reduced the space required for transmission lines. (This was followed by the development of coaxial cables, which could transmit hundreds of voice signals multiplexed onto a single cable.)

This evolved to large cables, "flexible as a sewer pipe," enclosing hundreds of twisted pairs plus several coaxial cables. Most of this cable, installed from about 1930 to date, is still in operation, mainly in metropolitan